ADMIN
Network & Security

Data Obfuscation | **Secure AD Design** | **Flipper Zero**

ADMIN
Network & Security

ISSUE 86

Rocky Linux™
Minimal 9.5
ISSUE 86/2025
ADMIN
Network & Security
DVD INSIDE

Data Obfuscation
Anonymize and pseudonymize your data

OpenFaaS
Deploy automated, optimized, and secure serverless applications

Secure AD Design

Tactical RMM
Open source remote monitoring and management

Pritunl Self-Hosted VPN Server

Passwords and Passphrases

OTOBO
An advanced ticketing and help desk system

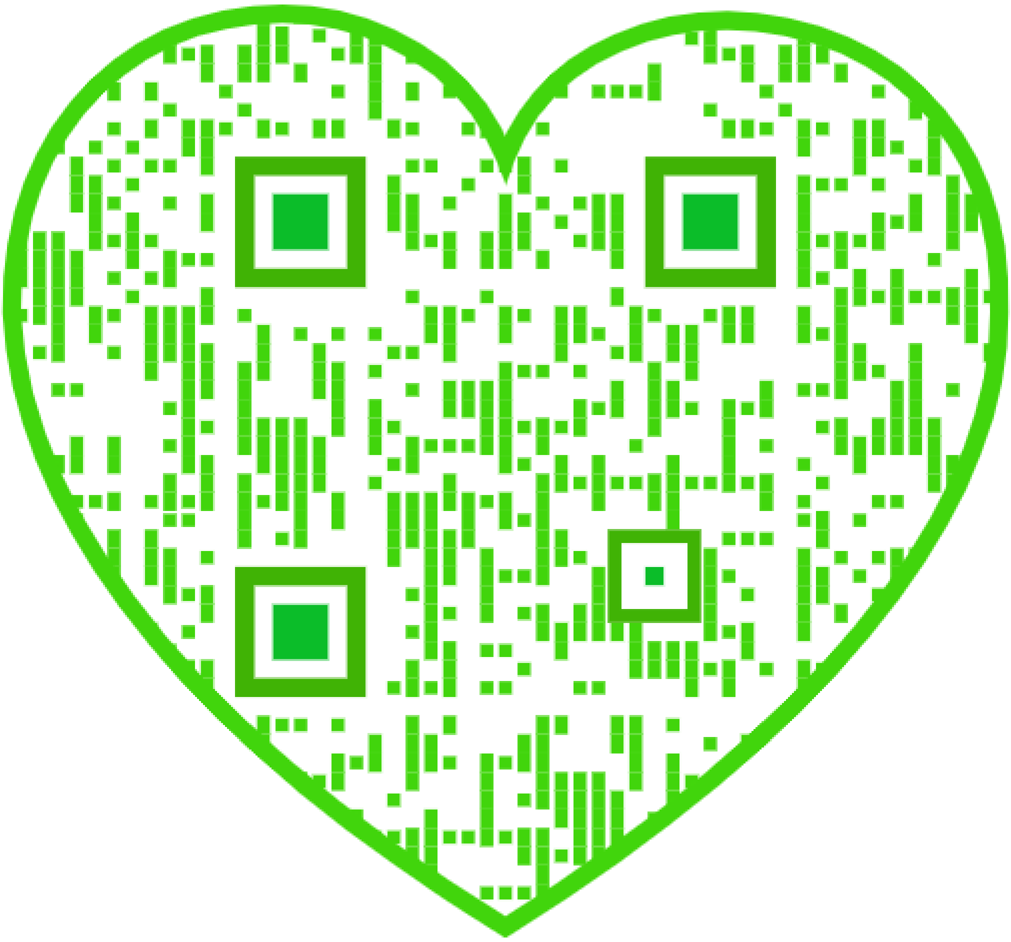
Flipper Zero
Discover vulnerabilities in infrastructures and configurations

Open-AudIT
Inventory, manage, and monitor IT assets

bootc Images
A container-based workflow creates bootable OS images

LINUX NEW MEDIA
The Pulse of Open Source

WWW.ADMIN-MAGAZINE.COM



Silent Computing Giant
TUXEDO PRO Desktop Series

The silent computer experience – the TUXEDO PRO Desktop Series is available in three different sizes and features a soundproofed aluminum-steel case. This ensures stability and whisper-quiet operation, while the integrated CPU water cooling keeps temperatures low. Customize your TUXEDO PRO to fit your exact needs – whether for gaming, creative projects, or professional applications. Your requirements take center stage. Plus, thanks to the wide range of connectivity options, your setup remains flexible and future-proof.

Linux compatible
Up to 5 Years Guarantee
Immediately ready for use
TUXEDO
tuxe.do/lxadmin86
Made in Germany
German Data Privacy
German Tech Support

The Vital Importance of Being Earnestly Vigilant

No matter what you do to secure and lock down your systems, your users can easily undermine you.

Security can seem like a comedy of errors, in that you have host-based firewalls, an intrusion detection system (IDS), an intrusion prevention system (IPS), security information and event management (SIEM), and mobile device management (MDM); you use network address translation (NAT), require a virtual private network (VPN) for remote connectivity, keep all your systems updated, keep an application safe list and a deny list, encrypt your in-flight and at-rest data, require frequent and complex password changes, read reams of CVEs, and track every possible vulnerability in the known galaxy. Yet, some unwitting user clicks on an email attachment that somehow made it through your spam filter and botches the entire network with malware that sends you and the rest of the IT staff scrambling for days.

Yes, that's the life of a system administrator in every contemporary office environment on the planet. These are indeed the times that try our souls.

You can implement 15 layers of security that are so elaborate and complex that the late Kevin Mitnick, infamous hacker and security consultant, would have declared your network certifiably secure. However, one person with happy fingers can expose your entire network to some faraway advanced persistent threat (APT) group who wants to steal your company's top-secret formulas and sell them to the dark web's highest bidder.

You might begin to wonder if you can keep your data, users, and systems safe when everyone is working against you, including your users. It's bad enough that bad actors constantly hammer, scan, and probe your external systems, searching for a way in, but the users you've trained and drilled on computer security participate in a collective *et tu, Brute?* moment that makes you question your sanity for choosing system administration as a career. It's OK. We've all been there. If your paranoia level moves from DEFCON 5 to DEFCON 2 at the mere hint of an insider threat, congratulations, you've joined the ranks of those who have met this challenge multiple times.

Those who have managed security and users for more than 20 years have learned to live with an unhealthy level of user disdain. These sys admins are more comfortable fighting APT groups, script kiddies, a variety of random folks with malicious intentions, and even insiders who want to harm, but what they're not comfortable with is the constant battle with those whose only defense is "I didn't think it was dangerous" or "I didn't change anything." Crossing security swords with invisible enemies is challenging, but doing so with someone who could be your parent or grandparent is impossible. You can be frustrated with them, but you must restrict your anger and take that condescending edge out of your voice when explaining the problem.

Yes, it's true. Your greatest security threat is the insider who clicks the link, plays the game, responds to the email, calls the 800 number from a text, or logs into the website that's off by one letter and reveals corporate information, installs malware, or is social engineered into providing access to your internal systems. These threats turn your hair as white as the sands of White Sands National Park in New Mexico. Find photos of yours truly for reference.

Training doesn't help. A security consulting company provided training to my users that included drills, examples, scenarios, and proper responses to all types of threats. Just two days later, the trainer baited us with a bogus email link that almost 50 percent of the freshly trained staff gladly clicked, proving my point.

I'm sorry to report that there is no solution unless it's some artificial intelligence bot that monitors a person's actions and prevents them from clicking, answering questions, responding to bogus surveys, or picking up the phone and carrying on a conversation. This fact doesn't mean you shouldn't try, but just take a reality check and save your sanity by accepting the truth that your best efforts are ignored, forgotten, or taken too lightly.

Constant reminders, short training sessions, and drills are your only hope in avoiding a few incidents. Remember the vital importance of being earnestly vigilant.

Ken Hess • Senior ADMIN Editor

Lead Image © archman, 123RF.com

ADMIN
Network & Security

Features

10 Data Obfuscation Techniques
Data anonymization and pseudonymization are two key techniques that ensure privacy while enabling the use of data for analysis and decision making; two methods offer different approaches that vary as a function of use case.

Service

3 Welcome
6 News
96 Back Issues
97 Call for Papers
98 Coming Next Month

Tools

16 Self-Hosted Chat Servers
Chat servers likely send confidential data to the cloud, so if you don't want to see your business on Google or Amazon, you can keep your information safely in your own data center with self-hosted options.

22 TutaCrypt
Your encrypted email messages might not be so secret after the quantum revolution picks up steam. Tuta Mail comes with a quantum-resistant alternative for users who take the long view.

26 OpenFaaS
Use advanced techniques for automation, optimization, and security to deploy serverless applications in a Linux environment.

30 Volcano
Optimize high-performance workloads on Kubernetes to avoid deadlocks.

36 Pritunl VPN Server
We offer the ultimate guide to deploying a self-hosted Pritunl VPN server with multifactor authentication.

Containers and Virtualization

42 Podman Desktop
Podman has become an attractive alternative to Docker, not least because it does not require a central daemon. We look at how to set about changing the guard and how you can use Podman Desktop to do so.

48 bootc Images
Use a container-based workflow to create bootable operating system images. In a few simple steps, you can use Podman to create ready-to-run images.

Security

50 Flipper Zero
This mobile hacking tool can check a wide range of wireless connections for vulnerabilities. We look at potential applications in pen-testing scenarios.

54 Maltrail
Established sources form the basis of this traffic analysis software to identify malicious traffic on your network.

58 Slowloris
Harden your web servers against denial-of-service attacks that restrict the availability of web services.

@adminmag

@admin-magazine.com

ADMIN magazine

@adminmagazine@hachyderm.io

10 | Data Obfuscation

Anonymization and pseudonymization of data

Generalization, suppression, perturbation, and differential privacy are essential data protection techniques that enable a balance between data security and usability and ensure compliance with legal requirements.

Highlights

22 TutaCrypt
Mitigate the risk of quantum attacks with quantum-resistant algorithms that ensure data remains secure now, as well as in the future.

50 Flipper Zero
A practical, multifunctional, mobile hacking multitool for pen testers that can be used to check a variety of devices and services for potential configuration errors.

72 Tactical RMM
This state-of-the-art tool for remote monitoring and management has a wide-ranging feature set with high performance, low maintenance overhead, and comprehensive reporting.

Management

62 Open-AudIT
Inventory, manage, and monitor IT assets, whether hardware, software, network device, or virtual machine. We explain how to install the software and work through the initial configuration.

66 OTOBO
OTOBO is a user-friendly and completely free ticket system designed to handle complex service requirements, offering an alternative, easy-to-use support tool with Active Directory integration and single sign-on.

72 Tactical RMM
This open source remote monitoring and management solution has agents to support Linux, macOS, and Windows.

Nuts and Bolts

78 Managing Dotfiles
The GNU Stow tool combined with Ansible can help you manage configuration files.

80 Passwords and Passphrases
Rules for secure passwords pose major obstacles for unsuspecting users, making passphrases and passkeys more important. Two recommendations illustrate team-capable password management.

86 Secure AD Design
Active Directory can be particularly problematic in terms of security. We reflect on the secure design, rigorous hardening, and security-conscious management of this service.

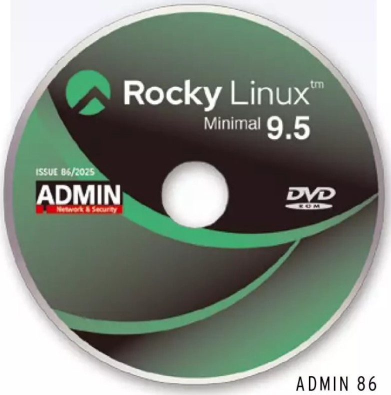
92 New sudo Log Format
An example shows how syslog-ng uses the new logging format of the sudo tool to process sudo messages.

On the DVD

Rocky Linux 9.5 Minimal
Rocky Linux is a downstream community-supported, binary-compatible release of Red Hat Enterprise Linux (RHEL). The minimal install offered here has the packages needed for basic functionality. Note that "Rocky Linux does not offer an upgrade path from any version of Rocky Linux 8. We recommend doing a fresh OS install to move to Rocky Linux 9.5" [1]. Some older CPUs will not work well or at all. You can find instructions for how to test your CPU online [2].

[1] Release notes: https://docs.rocky-linux.org/release_notes/9_5/

[2] Compatibility: https://docs.rockylinux.org/gemstones/test_cpu_compat/



News for Admins

Tech News

Palo Alto Networks Introduces Cortex Cloud

Palo Alto Networks has released Cortex Cloud, which merges Prisma Cloud with Cortex CDR.

This new version offers cloud detection and response capabilities along with AI-powered prioritization, automated remediation, and a simplified user experience.

According to the announcement, “Cortex Cloud identifies and prioritizes issues across the entire development pipeline with end-to-end context across code, runtime, cloud and newly introduced third-party scanners.”

“Cortex Cloud identifies high-risk threats with precision using Cortex XDR’s best-in-class CDR agent, which achieved industry-leading results in the most recent MITRE ATT&CK testing,” says Elad Koren (https://www.paloaltonetworks.com/blog/2025/02/announcing-innovations-cortex-cloud/), VP of Product Management, Prisma Cloud in a related blog post.

Read more at Palo Alto Networks: https://www.paloaltonetworks.com/cortex/cloud.

Canonical to Provide 12 Years of Kubernetes Support

Canonical is committing to 12 years of security maintenance and support, starting with Canonical Kubernetes 1.32.

“Canonical’s Kubernetes LTS (Long Term Support) will support FedRAMP compliance and receive at least 12 years of committed security maintenance and enterprise support on bare metal, public clouds, OpenStack, Canonical MicroCloud and VMware,” the announcement states.

“Constant Kubernetes upgrades are a drain on enterprise teams. Customers who deploy Canonical Kubernetes 1.32 LTS can focus on the future, because their clusters will receive security updates for 12 full years,” Mark Shuttleworth said.

Canonical Kubernetes 1.32 LTS is currently available and ready for production use. Read more at Canonical: https://canonical.com/blog/12-year-lts-for-kubernetes.

Mirantis Releases Open Source k0rdent

The Mirantis team has released k0rdent, a fully open source Distributed Container Management Environment (DCME), which provides a single point of control for cloud-native applications on any infrastructure.

According to the announcement, k0rdent’s job is “to holistically provision and manage the diverse underlying infrastructure (clouds, bare metal, edge devices) needed to build developer and workload platforms at scale.”

The Kubernetes-native tool helps you:

- Deploy Kubernetes anywhere with consistent automation and policy enforcement.
- Provision and manage core services consistently and automatically.
- Provide self-service, composable infrastructure.
- Scale modern workloads seamlessly.
- Enforce security and compliance at scale.

Learn more at k0rdent: https://k0rdent.io/.

Get the latest IT and HPC news in your inbox

Subscribe free to ADMIN Update

bit.ly/ADMIN-Update

Lead Image © vlastas, 123RF.com

D-Wave Now Offers On-Premises Quantum Computing Systems

Research centers, academic institutions, and governments can now purchase on-premises D-Wave Advantage quantum computing systems, according to a recent announcement from D-Wave.

D-Wave Advantage systems are “the world’s largest annealing quantum computers, with more than 5,000 qubits and 15-way connectivity,” the company says. And, with on-premises installation, customers can “tightly integrate with existing classical and HPC systems, leverage and tune system parameters, and take advantage of new system innovations, such as advanced analog-digital features, as they are developed.”

“By integrating annealing quantum computing with HPC, I predict we will see significant progress in quantum research and quantum AI development, leading to new discoveries and improved outcomes,” said Dr. Alan Baratz, CEO of D-Wave.

In conjunction with the news, the company announced the first purchase of a D-Wave Advantage quantum system by the Forschungszentrum Jülich HPC center (https://www.fz-juelich.de/en/news/archive/press-release/2025/forschungszentrum-julich-strengthens-quantum-research-with-purchase-of-d-wavetm).

Read more at D-Wave: https://www.dwavesys.com/solutions-and-products/systems/.

IP Fabric 7.0 Released

IP Fabric has announced the 7.0 release of its Automated Network Assurance Platform (https://ipfabric.io/how-does-it-work/), which aims to provide the necessary insight and control to manage complex cloud, security, and network environments.

Specifically, IP Fabric uncovers critical details about your infrastructure, “including asset inventory, state, end-to-end paths, configurations, policy application, dependencies, segmentation and more,” says Kristine Botha. This visibility can help reveal risks such as unmanaged devices, compliance issues, and security vulnerabilities.

New features in IP Fabric 7.0 include:

- Multi-view dashboards for tracking security operations or monitoring compliance.
- Visio export for creating detailed diagrams.
- Expanded security capabilities including Check Point, Palo Alto, and Stormshield support.
- New Border Gateway Protocol (BGP) detail.

Check out the demo and learn more at the website: https://ipfabric.io/.

UK Releases Code of Practice for Securing AI

The UK government has developed a voluntary Code of Practice aimed at addressing AI cybersecurity risks.

This Code of Practice applies to developers, system operators, and organizations that create, deploy, or manage AI systems. And, according to the announcement, it “equips organizations with the tools they need to thrive in the age of AI. From securing AI systems against hacking and sabotage, to ensuring they are developed and deployed in a secure way, the Code will help developers build secure, innovative AI products.”

Specifically, the Code sets out 13 cybersecurity principles encompassing the software development lifecycle — secure design, secure development, secure deployment, secure maintenance, and secure end of life. The general principles are:

1. Raise awareness of AI security threats and risks.
2. Design your AI system for security as well as functionality and performance.
3. Evaluate the threats and manage the risks to your AI system.
4. Enable human responsibility for AI systems.
5. Identify, track and protect your assets
6. Secure your infrastructure.
7. Secure your supply chain.
8. Document your data, models and prompts.
9. Conduct appropriate testing and evaluation.
10. Communication and processes associated with end-users and affected entities.
11. Maintain regular security updates, patches and mitigations.
12. Monitor your system’s behavior.

13. Ensure proper data and model disposal.

See the announcement for details: https://www.gov.uk/government/publications/ai-cyber-security-code-of-practice/code-of-practice-for-the-cyber-security-of-ai#introduction.

Red Hat Releases Kubernetes-Native Connectivity Link

Red Hat has released Red Hat Connectivity Link, aimed at reducing complexity and providing a single solution for connecting applications and infrastructure.

The Kubernetes-native solution, which is based on the open source Kuadrant project (https://kuadrant.io/) and builds on the Kubernetes Gateway API (https://gateway-api.sigs.k8s.io/), "ensures seamless traffic management and policy enforcement in a single cluster or across multiple clusters," the website says.

Red Hat Connectivity Link also enables features such as dynamic DNS and multi-cluster routing, automatic TLS integration, controlled access, rate limiting for ingress traffic, and observability.

Read more at Red Hat: https://developers.redhat.com/products/red-hat-connectivity-link/overview.

OpenVox Automation Framework Announced

The Vox Pupuli team has announced the first release of OpenVox, a community-maintained, open source implementation of the Puppet configuration management platform. According to the announcement, OpenVox 8.11 is functionally equivalent to Puppet and should be considered a drop-in replacement.

"The command names are the same, the configuration file paths are the same, etc. The major differences are in help text output, man pages, and so on," the install page (https://voxpupuli.org/openvox/install/) says. "This means that you can continue to use all the commands, modules, tooling, etc. that you're used to, but at this time you cannot install both Puppet and OpenVox on the same system."

OpenVox is considered a soft fork, the team says, because they intend to maintain downstream compatibility for as long as possible. "As such, Vox Pupuli is working with Perforce to create a Puppet Standards Steering Committee to set the direction of features and language evolutions."

Learn more at OpenVox: https://voxpupuli.org/openvox/.

Sysdig Launches Open Source Stratoshark for Cloud Observability

Sysdig has announced Stratoshark, an open source tool that extends Wireshark's network visibility into the cloud.

Wireshark (https://www.wireshark.org/) has helped users analyze network traffic and troubleshoot issues for 27 years, the announcement says. Now, Stratoshark (described as "Wireshark for the Cloud") offers deep cloud observability to help admins analyze and troubleshoot cloud system calls and logs using familiar functionality.

The tool combines packet analysis with Falco's runtime security (https://sysdig.com/opensource/falco/) to help teams accelerate incident response and adapt to the rapidly changing demands of cloud-native systems.

Read more at Sysdig: https://sysdig.com/opensource/stratoshark/.

Data Center Electricity Demand Projected to Double or Triple by 2028

Data center electricity demand in the United States has tripled in the past decade and is projected to double or triple again by 2028, according to the 2024 United States Data Center Energy Usage Report (https://eta-publications.lbl.gov/sites/default/files/2024-12/lbnl-2024-united-states-data-center-energy-usage-report.pdf)from the Department of Energy (DOE).

The report examines historical data center electricity consumption back to 2014 and provides a range of scenarios for demand out to 2028. It also looks at cryptocurrency electricity use, server types and associated power draw, and water and emission impacts.

According to the report, US data center annual energy use remained stable in 2014–2016 at about 60 terawatt hours (TWh) and began to increase in 2017 as GPU-accelerated servers became a more significant part of data center server stock.

- By 2018, data centers consumed about 76 TWh, representing 1.9 percent of total annual US electricity consumption.
- By 2023, energy use reached 176 TWh, or about 4.4 percent of total consumption.
- The report estimates an increase to between 325 and 580 TWh by 2028.

The projected use estimates consider factors such as data center expansion, an increase in AI applications, domestic manufacturing growth, and more.

The report also details direct and indirect data center water usage and greenhouse gas emissions, noting that "with the projected growth of data centers' energy use in the coming years, indirect water consumption and emissions are also expected to increase."

Read the complete report at the Department of Energy: https://www.energy.gov/articles/doe-releases-new-report-evaluating-increase-electricity-demand-data-centers.

Red Hat Releases OpenShift Virtualization Tools

Red Hat has announced the general availability of Red Hat OpenShift Virtualization Engine, which is focused on virtualization workloads, along with Red Hat Advanced Cluster Management for Virtualization (https://app.arcade.software/share/PIbyWf2YD2A6BqwXp6hz), aimed at unifying virtual machine (VM) management at scale.

The OpenShift Virtualization Engine simplifies the process of deploying, managing, and scaling VMs, with key features including:

- Bare-metal deployment, supporting up to 128 cores per bare-metal socket pair, which is double the previous amount.
- Over-the-air updates.
- Dedicated virtualization administrator view within the web-based management UI.
- Automated provisioning of VMs with OpenShift GitOps.
- Enhanced VM platform logging.

Additionally, Red Hat Advanced Cluster Management for Virtualization provides "a centralized platform for managing virtual machines throughout their lifecycle with consistency and compliance," the website says. The tool integrates with the Ansible Automation Platform for improved automation and efficiency.

Learn more at Red Hat: https://www.redhat.com/en/blog/introducing-red-hat-openshift-virtualization-engine.

16GB Raspberry Pi 5 Announced

The Raspberry Pi Foundation has released (https://www.raspberrypi.com/news/16gb-raspberry-pi-5-on-sale-now-at-120/) a 16GB variant of the Raspberry Pi 5, joining the 2GB, 4GB, and 8GB offerings released previously.

According to the website, the Raspberry Pi 5 "features the Broadcom BCM2712 quad-core Arm Cortex A76 processor @ 2.4GHz, making it up to three times faster than the previous generation. With RAM variants up to 16GB, this is the fastest, smoothest Raspberry Pi experience yet."

This latest version, intended for power users and professional applications, is available now for $120.

Read more at Raspberry Pi: https://www.raspberrypi.com/products/raspberry-pi-5/.

Open Storage Network Adds More Sites

The Open Storage Network (OSN), which offers low-cost, sustainable distributed data storage for the research community, is adding storage nodes in the United States to expand the network and address growing needs.

The OSN, which was established in 2018, "supports allocations of up to 50 terabytes at no charge to researchers via ACCESS (https://access-ci.org/), as well as paid participation for entities with larger needs," the announcement (https://access-ci.org/open-storage-network-welcomes-new-campus-computing-partners/) says.

"We built the Open Storage Network on the premise that a collaborative effort, rooted in the research computing community, could address an acute need for simple and cost-efficient storage and sharing of scientific data. Results so far have exceeded expectations," said Alex Szalay, founder of the OSN project.

You can get an allocation (https://www.openstoragenetwork.org/get-involved/get-an-allocation/) or learn how to host a pod (https://www.openstoragenetwork.org/get-involved/get-a-pod/) at the OSN.

Anonymization and pseudonymization of data

Behind the Mask

Data anonymization and pseudonymization are two key techniques that ensure privacy while enabling the use of data for analysis and decision making; the two methods offer different approaches that vary as a function of use case. By Daniel Richey

Data anonymization and pseudonymization are not just technical approaches but should be key factors in every organization's data protection strategy. Whereas anonymization aims to modify personal data such that the data subject cannot be identified, pseudonymization aims to make identification more difficult by replacing identifiers with pseudonyms. These techniques are particularly relevant in the context of Europe's General Data Protection Regulation (GDPR), which stipulates strict requirements for the handling of personal data, requiring businesses to implement robust mechanisms that ensure both data integrity and data protection.

Implementing anonymization and pseudonymization techniques is technically demanding and poses numerous challenges, including ensuring data quality and usefulness after anonymization and providing protection against re-identification risks. At the same time, these techniques offer immense opportunities, particularly with a view to leveraging big data and machine learning without violating data protection regulations. Choosing the right technology and implementing it requires a deep understanding of the existing data structures, the legal framework, and the potential risks. Various technical processes play a role and are discussed in detail in this article.

Different Ways for Different Plays

Anonymization and pseudonymization differ fundamentally in their application and objectives and depend on the industry and use case. Anonymization involves modifying personal data to remove identifiability irreversibly. Direct and indirect identifiers are removed or modified. After complete anonymization, it is no longer possible to draw conclusions about individual people, which puts the data firmly outside the scope of the strict requirements that apply under data protection law. Once anonymized, data can no

Photo by Debby Ledet on Unsplash

longer be traced back to the original subject, which makes this method ideal for open data, research, and analysis where conclusions about specific people are not needed. In healthcare, for example, it is essential to provide anonymized patient data for research purposes without compromising patient privacy.

Pseudonymization, on the other hand, means processing personal data such that it can no longer be mapped to a specific person, while retaining the ability to re-identify under certain circumstances. Direct identifiers are replaced by pseudonyms, and additional information (e.g., keys or reference tables) is stored separately to protect it through technical and organizational measures. This method is particularly suitable for use cases in which subsequent identification might be necessary. In the financial sector, pseudonymization is often used for transaction data analysis and fraud detection without revealing the identity of the customer.

Anonymization Techniques

Regulations such as the GDPR make it clear how important it is to comply with legal requirements for anonymization and pseudonymization. Articles 25 and 32 of the GDPR require companies to implement appropriate measures to anonymize or pseudonymize personal data to protect the rights and freedoms of data subjects.

Various techniques are available for data anonymization that vary as a function of the use case and the specific requirements. I look at some of the most important anonymization techniques in detail, including their advantages and disadvantages.

Generalization is a technique in which you group specific data into less precise but still informative categories. This method reduces the risk of re-identification by reducing the granularity of the data. For example, instead of specifying a person's exact age, you could specify an age group (e.g., 30-40 years of age). This approach preserves the usefulness of the data for statistical analysis while reducing identifiability. However, generalization can affect the precision and detail of the data, which has disadvantages in some analysis scenarios.

Suppression, on the other hand, means removing or masking individual data fields completely to prevent identifiability. This method is particularly useful if certain attributes are very specific and therefore identifiable. For example, you can replace the last four digits of an ID with *XXXX*. This technique is easy to use and effective in protecting privacy. However, it can significantly reduce the usefulness of the data if too much information is suppressed.

Perturbation takes a different tack, changing the original data by adding noise or random modifications (Figure 1). This method ensures that individual data points can no longer be traced back to the original values. One example of perturbation is adding random deviations to geolocation data, which preserves the overall structure of the data and still enables data scientists to carry out aggregated analyses while the individual data points remain anonymous. One disadvantage of perturbation is that it can reduce the accuracy of the data, which can be particularly problematic in scenarios that require precise analysis.

Differential privacy is an advanced technique that enables data scientists to carry out statistical analyses on anonymized datasets without increasing the risk of re-identification. This method adds controlled noise to the data to minimize the traceability of individual datasets. One example of this method is the publication of

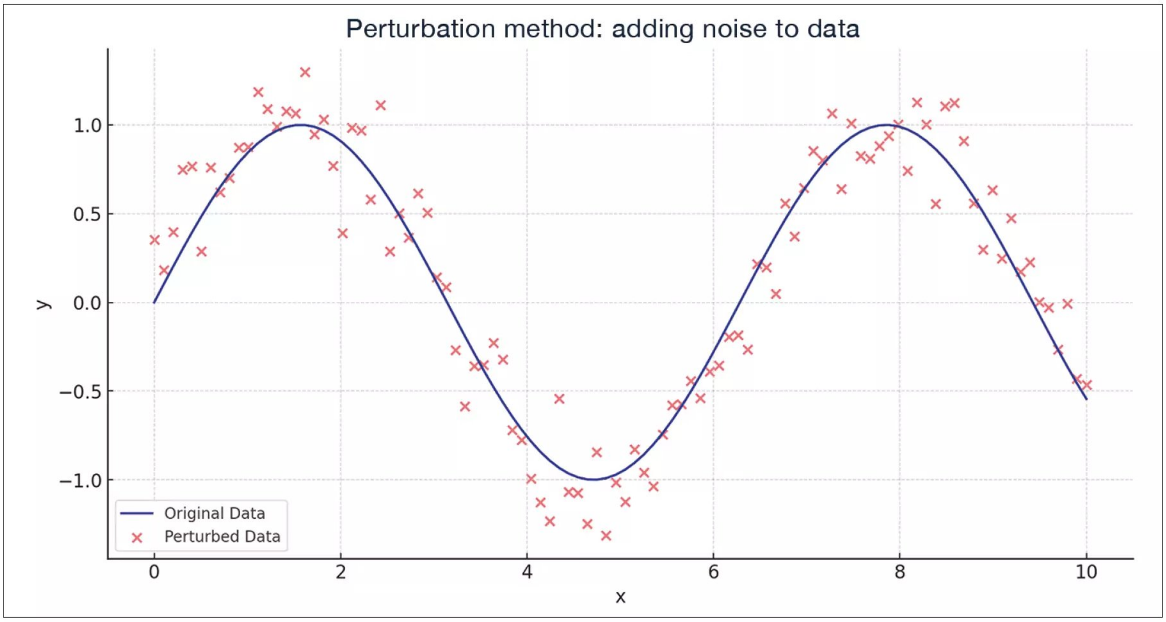


Figure 1: **A perturbation: The blue line represents the original data, whereas the red dots represent the modified data created by adding noise.**

Table 1: k-Anonymity

ID	Age	Gender	Zip Code
Original Data			
1	34	M	12345
2	36	F	12346
3	37	F	12347
4	35	M	12345
5	36	F	12346
6	34	M	12347
Anonymized Data (k=3)			
1	30-40	*	123##
2	30-40	*	123##
3	30-40	*	123##
4	30-40	*	123##
5	30-40	*	123##
6	30-40	*	123##

aggregated data, where random modifications are added to maintain the accuracy of the overall results. This approach is particularly suitable for scenarios in which you need to perform an accurate statistical analysis without jeopardizing the privacy of individuals. The main drawbacks are the complexity of its implementation and the need to find the right level of noise to ensure both privacy and data quality.

K-anonymity, on the other hand, is a strategy for ensuring that each data record in an anonymized database cannot be distinguished from less than *k* – 1 other data records. You can achieve *k*-anonymity through generalization and suppression, so that every combination of attributes occurs in at least *k* data records. One classic example of this method is generalizing zip codes to the first three digits to ensure that a large group of people share the same zip code (Table 1). The advantage of *k*-anonymity is its ease of use and the protection it provides against re-identification. However, it can lead to a loss of information and reduced data quality if the generalization is not sufficiently granular.

L-diversity extends *k*-anonymity to ensure a sufficient diversity of sensitive attributes within each group of *k* data records. This method reduces the risk that confidential information can be derived from the data. *t*-closeness goes one step further and ensures that the distribution of sensitive attributes in each group is similar to the distribution in the population as a whole. These techniques provide additional layers of protection by preventing identifiability and making it impossible to derive sensitive information. The advantage of these methods lies in their robust protection against inference attacks, whereas disadvantages include complexity and the increased overhead required for data processing.

Masks On!

Pseudonymization allows the original data to be reconstituted with additional information according to a number of methods.

Tokenization is a technique in which sensitive data is replaced with meaningless tokens. These tokens have no inherent value and do not allow any conclusions to be drawn about the original data. For example, you can replace credit card numbers with random character strings. The main advantage of tokenization is its simplicity and effectiveness. Tokens can be easily used in most IT systems without the need for extensive changes. Another advantage is that you can track the tokens and reassign them to the original data if required, provided the token database is managed securely. One disadvantage of tokenization, however, is that it relies on secure management of the token database. If this database is compromised, it will be possible to draw conclusions about the original data.

Encryption protects data by cryptography and ensures that only authorized persons with the matching key can access the original data. The advantage of encryption lies in its high level of security. Even if the data is sniffed, it is unreadable without the matching key. Another advantage is encryption's flexibility, which makes it suitable for different types of data and applications. The disadvantage is the complexity that key management entails. The keys must be stored and managed securely, which requires additional overhead and resources.

Masking replaces parts of the data with fixed characters to prevent readability and is particularly useful in development or test environments when real data is not required. For example, you can partially mask phone numbers by displaying only the first or last digits (e.g., 171-XXXXXXX). The advantage of masking is its simplicity and its ease of implementation without changing the data structure. Masked data is usually fine for test and development purposes. One of its disadvantages is that it does not offer complete security. Masked information can be partially recoverable through pattern recognition or other techniques.

Hashing describes a technique that involves converting data into a fixed character string by a hash function. This character string, also known as a hash, represents the original data but cannot be easily converted back. One common example of hashing is the SHA-256 cryptographic hash function. The advantage of this method lies in its simplicity and efficiency. Hashes are static and easy to compare, which makes them ideal for applications such as password storage. Another advantage is the immutability of the hash values, which ensures that even minor changes to the original data results in completely different hashes. One drawback with hashing is that it can be susceptible to collisions and rainbow table attacks, especially if insecure or weak hash functions are used. To prevent this attack, it makes sense to use salting, which injects additional random data into the hashing process.

Challenges and Risks

Implementing data anonymization and pseudonymization techniques in an organization comes with specific challenges and risks that you need to consider as an admin. These methods not only require technical expertise, but also strategic planning and an understanding of the potential pitfalls. Complexity of implementation is one of the biggest hurdles. Especially in small and medium-sized enterprises

(SMEs), which often do not have extensive IT resources, selecting and integrating suitable anonymization and pseudonymization techniques is a challenging task. You need to ensure that the selected methods integrate seamlessly with existing systems without compromising data consistency or system performance.

Another technical problem lies in achieving a balance between data protection and data quality. Anonymization techniques such as generalization and perturbation can significantly reduce the usefulness of the information. You need to consider carefully the extent to which information is modified to ensure that both data is protected and values are usable for analysis.

Integration with existing IT infrastructures can also be difficult. Many systems are not designed to process anonymized or pseudonymized data, which requires adjustments and possibly significant modifications. You also need to take care that all systems and applications involved communicate correctly with each other and have no vulnerabilities.

Re-identification risks are omnipresent despite careful anonymization or pseudonymization. Attackers can try to combine anonymized data with other datasets to identify individuals. To mitigate this risk, the use of advanced techniques such as differential privacy and regular security checks make sense.

Managing keys and tokens is crucial in pseudonymization. Compromises involving this security information can lead to re-identification of the data, which makes strong encryption procedures and secure key management systems essential. Regularly checking and updating security protocols should be a matter of course.

Human error also poses a risk. Employees without sufficient training could introduce vulnerabilities or use insecure practices that undermine anonymization and pseudonymization measures. This danger makes regular training and awareness programs crucial to mitigate the risk of human error.

Useful Tools

The practical implementation of anonymization and pseudonymization techniques requires the use of specialized tools and compliance with best practices (see the "Best Practices" box).

Apache Spark **[1]** is a powerful open source engine for large-scale data processing. With the help of the PySpark Python interface, you can use Spark for data anonymization and pseudonymization. Spark offers extensive libraries for data manipulation and can process large amounts of data efficiently. For example, you can apply generalization and perturbation techniques to large datasets to anonymize personal data. PySpark lets you implement complex anonymization algorithms in a scalable environment.

ARX **[2]** is an open source tool for anonymizing data. It supports various anonymization techniques such as *k*-anonymity, *l*-diversity, and *t*-closeness. With the help of ARX, you can explore, analyze, and transform data to ensure compliance with data protection requirements. The tool offers a graphical user interface and an API that you can integrate into your own applications. This tool is particularly suitable for detailed anonymization projects and offers a wide range of configuration options.

sdcMicro **[3]** is an R package for statistical discretization and anonymization of microdata (**Figure 2**). It offers functions for applying *k*-anonymity, *l*-diversity, and other anonymization techniques. sdcMicro lets you anonymize data directly in R, making it useful for statistical analysis. The package can create anonymization reports and comes with anonymization quality assessment tools to ensure that the data remains both secure and analytically usable.

The *format-preserving encryption (FPE)* **[4]** technique preserves the structure of the original data. FPE is particularly suitable for the pseudonymization of data such as credit card numbers or social security numbers, because the encrypted values have the same format as the original values. Tools such as the NIST-standardized FF1 and FF3 methods provide FPE implementations that you can integrate into your applications to pseudonymize data securely.

Built-in functions for encrypting and pseudonymizing data come with modern databases such as PostgreSQL **[5]**, MongoDB **[6]**, and Cassandra **[7]**. For example, you can use PostgreSQL

Best Practices

- Before employing anonymization and pseudonymization techniques, it makes sense to carry out thorough data classification and evaluation by identifying sensitive data and assessing the risk of re-identification. A clear-cut classification will help you to select and apply the appropriate anonymization and pseudonymization techniques.
- Use salts when hashing to ensure additional security. A salt adds random data to each record before it is hashed, minimizing the risk of rainbow table attacks. This technique significantly improves the security of hashed data and should be used by default.
- Implement continuous monitoring and evaluation processes to ensure the effectiveness of your anonymization and pseudonymization measures. Regular audits and penetration tests can help you identify and eliminate potential vulnerabilities. Continuous reviews of your data protection strategies ensure that they comply with current threats and legal requirements.
- Document all anonymization and pseudonymization processes transparently and in detail. This step not only helps with internal audits, but also when communicating with stakeholders and supervisory authorities. Clear and transparent documentation bolsters confidence in your data protection approach and makes it easier to comply with legal requirements.
- Use a combination of different anonymization and pseudonymization techniques to achieve a higher level of protection. You can increase security barriers and further reduce the risk of re-identification by applying multilayer security strategies. This approach ensures more robust protection and improves the overall security of your data processing operations.

extensions such as `pgcrypto` to encrypt data directly in the database **[8]**. MongoDB offers integrated encryption functions and enables the management of keys and tokens. These functions facilitate the implementation of data protection measures directly at the database level and offer high performance and security levels.

A Look into the Future

In recent years, differential privacy has established itself as a leading technique for ensuring data protection in large volumes of data. It allows data scientists to perform statistical analyses without jeopardizing the privacy of individual datasets.
In the future, increasing numbers of organizations are likely to implement differential privacy, especially in areas such as the public sector, healthcare, and financial services. Understanding and applying the principles and implementation details of differential privacy is important to remain competitive in these areas.
Blockchain technology could also play a key role in the future of data anonymization and pseudonymization. By decentralizing and cryptographically securing data, blockchain offers a robust approach to data processing and storage. Anonymization techniques can be integrated directly into blockchain protocols to ensure that transactions and data records are anonymized and immutable. Closely monitoring the development of blockchain technologies and their application in the field of data protection is a good idea.

Conclusions

Data anonymization and pseudonymization are essential data protection techniques that are particularly crucial for SMEs. They enable a balance between data security and usability and ensure compliance with legal requirements, in particular the GDPR. With the use of techniques such as generalization, suppression, perturbation, and differential privacy, along with tools such as Apache Spark and ARX, companies can effectively protect personal data. At the same time, pseudonymization methods such as tokenization and encryption provide robust security measures. Future technologic developments, including artificial intelligence and blockchain, will continue to change the data protection landscape. ■

Info

[1] Apache Spark: [https://spark.apache.org]

[2] ARX: [https://arx.deidentifier.org/anonymization-tool/]

[3] sdcMicro: [https://sdctools.github.io/sdcMicro/articles/sdcMicro.html]

[4] Format-preserving encryption: [https://en.wikipedia.org/wiki/Format-preserving_encryption]

[5] PostgreSQL Anonymizer 1.0: [https://www.postgresql.org/about/news/postgresql-anonymizer-10-privacy-by-design-for-postgres-2452/]

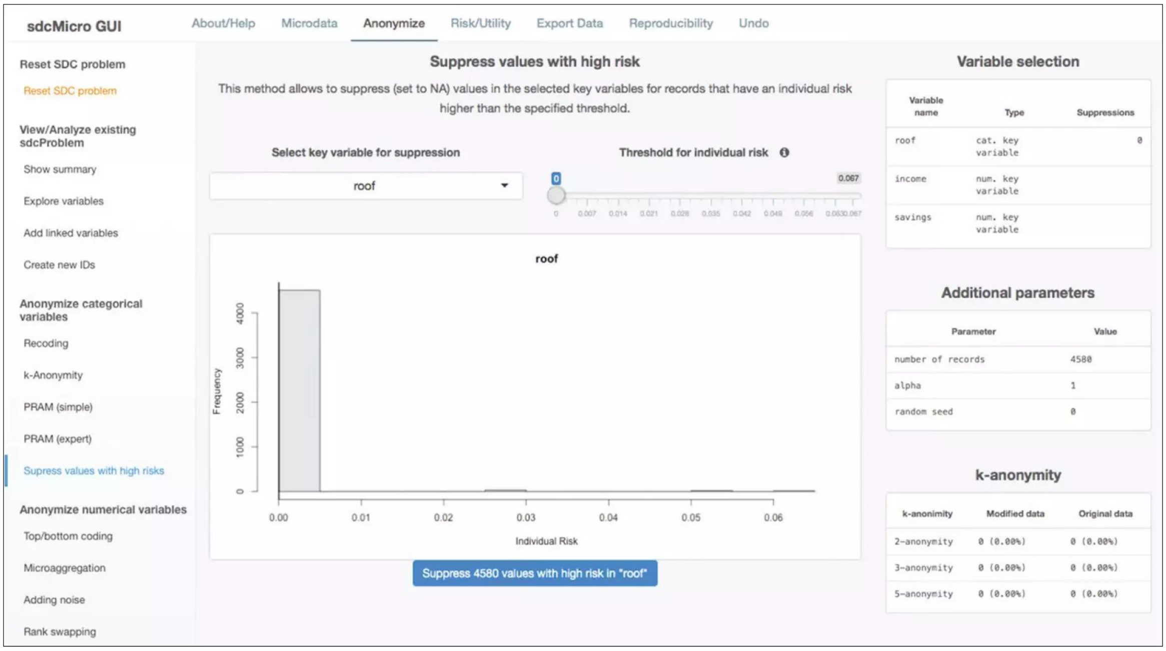
[6] MongoDB data masking: [https://www.mongodb.com/resources/products/capabilities/working-with-sensitive-data-while-keeping-it-secure-a-guide-to-data-masking]

[7] Cassandra DDM: [https://cassandra.apache.org/doc/latest/cassandra/developing/cql/dynamic-data-masking.html]

[8] pgcrypto: [https://www.postgresql.org/docs/current/pgcrypto.html]

The Author

Daniel Richey is deputy editor-in-chief at *IT-Administrator* magazine, where he covers the topics of IT security, infrastructure, and law. Before becoming an editor 20 years ago, Daniel studied economics and then taught economics and IT as a freelance lecturer at the Stuttgart University of Cooperative Education.


sdcMicro GUI
About/Help
Microdata
Anonymize
Risk/Utility
Export Data
Reproducibility
Undo
Reset SDC problem
Reset SDC problem
View/Analyze existing sdcProblem
Show summary
Explore variables
Add linked variables
Create new IDs
Anonymize categorical variables
Recoding
k-Anonymity
PRAM (simple)
PRAM (expert)
Supress values with high risks
Anonymize numerical variables
Top/bottom coding
Microaggregation
Adding noise
Rank swapping
Suppress values with high risk
Select key variable for suppression
roof
Threshold for individual risk
0
0.067
0
0.007
0.014
0.021
0.028
0.035
0.042
0.049
0.056
0.0630.067
roof
Frequency
0
1000
2000
3000
4000
0.00
0.01
0.02
0.03
0.04
0.05
0.06
Individual Risk
Suppress 4580 values with high risk in "roof"

This method allows to suppress (set to NA) values in the selected key variables for records that have an individual risk higher than the specified threshold.

Variable selection

Variable name	Type	Suppressions
roof	cat. key variable	0
income	num. key variable	
savings	num. key variable	

Additional parameters

Parameter	Value
number of records	4580
alpha	1
random seed	0

k-anonymity

k-anonimity	Modified data	Original data
2-anonymity	0 (0.00%)	0 (0.00%)
3-anonymity	0 (0.00%)	0 (0.00%)
5-anonymity	0 (0.00%)	0 (0.00%)

Figure 2: sdcMicro anonymizes microdata as an R package.

For the Community By the Community

PyConUS 2025

14 - 22 May 2025

Pittsburgh, PA

PyCon US is the largest annual gathering for the community
that uses and develops the open source Python programming language.

The core of the conference, May 16-18, packs in three days worth of
our community's best talks, amazing keynote speakers,
and our famed lightning talks to close out each day.
Beyond that, we'll have tutorials, sprints, social events,
and more for all levels and uses of Python.
We'd love for you to join this year's conference of all things Python– register today!

REGISTER AT
HTTPS://US.PYCON.ORG/2025/

SCAN ME

PYCON US 2025 IS A PRODUCTION OF THE PYTHON SOFTWARE FOUNDATION

Zulip, Mattermost, and Rocket.Chat

Close to the Chest

Chat servers likely send confidential data to the cloud, so if you don't want to see your business talks on Google or Amazon, you can keep your information safely in your own data center with self-hosted options – and without compromising on convenience.

By Andreas Stolzenberger

Spurred on by the huge success of tools such as WhatsApp and Telegram, increasing numbers of business users are now showing an interest in chat services as an alternative to email for internal company communication. Tools, bots, and plugins that integrate other applications directly into the chat forums are also useful. Office staff can discuss current business cases with coworkers in the team chat while a bot from the ticket system feeds information on currently open tickets and processing status.

Slack is currently one of the most popular chat services for businesses, partly because it integrates with more or less any commercially available business application and tool. However, as with all cloud applications, you must be careful: Confidential company communications end up somewhere in the Amazon cloud. Business users can choose the region in which they want to store their data and even which (other) country will store a backup, but basic organizational data and user profiles still end up on third-party servers.

Of course, you already know what's coming next at this point. If you love the convenience and functionality of tools like Slack but don't want to (or aren't allowed to) entrust your data to a cloud or a third party, you can always turn to self-hosted open source programs. In this article, I take a close look at the complicated setup and operation of the popular open source options Zulip [1], Mattermost [2], and Rocket.Chat [3] and discover how well these platforms integrate with other applications.

Functions and Features

A number of questions should be considered when looking for local chat groupware:

- Do clients exist for all required target platforms (web, desktop, mobile), and do they have push notifications?
- For which target platforms do ready-made plugins and bots exist?
- If no ready-made add-ons exist, how time-consuming is it to write your own and integrate them on the chat and application side?
- How secure is the data storage and backup of the platform?

Unlike earlier platforms such as the Internet Relay Chat (IRC) protocol or Network News Transfer Protocol (NNTP), modern chat groupware software no longer relies on proprietary binary protocols. All communication, including communication to the bots, relies on HTTPS. The chat server itself can hide behind an SSL-terminating proxy, which means that a chat environment does not require a dedicated IP address on the Internet, but runs on a hosted system or in the demilitarized zone (DMZ, a subnet that separates an internal network from the Internet) of the local data center behind a reverse proxy with name-based routing.

In security-relevant environments with an air gap (isolated network), self-hosted chat servers can also completely do without an Internet connection. They all work with the usual web application server architecture, involving a database server and a chat app, and the whole setup can be installed in the traditional way on a virtual machine (VM) or on a container platform such as Docker, Podman, or Kubernetes. I look at the advantages and disadvantages of each of these operating modes and try out various approaches.

Lead Image © rll, fotolia.com

Zulip

The Zulip chat server has been around longer than Slack, having first seen the light of day in 2012. The Zulip company offers both a software-as-a-service (SaaS) version in the cloud and commercial support for local setups. Although the community version is available at no charge, it also has no support. Technically and functionally, the commercial and the free versions are not different, and both have the full feature set.

Zulip requires a whole zoo of services on the back end. Redis, Memcached, and RabbitMQ team up with a PostgreSQL server for data storage. Memcached improves performance, especially in environments with many users, but Redis and RabbitMQ offer similar functionality, so choosing one of the two should be fine. The developers describe in the documentation why they decided not to consolidate the two services. The slightly more complex setup – especially in larger environments – is intended to distribute the load more evenly across services and improve performance.

Zulip can be set up along the classic approach (e.g., with a VM), although the vendor prefers an installation with containers and Docker. In the test environment for this article, I used a Linux, Ansible, NGINX, and Podman (LANP) setup, as described previously **[4]**. The Podman setup groups all the components as containers in a single pod and only shares one HTTP port with the outside world. The upstream NGINX reverse proxy takes care of SSL termination.

In principle, you just need to convert the Docker Compose file **[5]** provided by the vendor into an Ansible playbook (which I do not print here because of its sheer length), but with two changes: (1) Memcached tries to create a temporary directory in the current container, which will work in Docker with root privileges, but not in Podman; therefore, the Memcached container needs a persistent volume for the `/home/memcache` directory to run in Podman. (2) For Zulip to use HTTP behind a reverse proxy, you will need to edit the configuration file, because in the basic setup, the software tries to control the HTTPS protocol and integrate the certificates itself. Unfortunately, the Compose file does not envisage the Zulip configuration file being stored on a persistent volume. You need to add this option in the Ansible playbook:

```
volumes:
  - "/var/pods/zulip/zulip:/data:Z"
  - "/var/pods/zulip/etc:/etc/zulip:Z"
```

You can modify the `zulip.conf` configuration file in the host subdirectory (`/var/pods/zulip/etc`) to match by adding the lines

```
[application_server]
http_only = true
```

to transfer SSL termination to NGINX. You can then also customize the `settings.py` and `zulip-secrets.conf` configuration files to suit your organization's needs and to give Zulip access to a mail server. The software needs to be able to use SNMP to send messages with invitations for users. If you extend the mail setup to include an IMAP configuration, Zulip can receive email on its own addresses and add them to chat forums.

Zulip offers clients for all the popular client operating systems and mobile devices and runs in a browser, as well. In fact, all of the clients are basically just wrappers for a browser. The user interface (UI) in version 8.4 is tidy but looks a little old-fashioned with its 2010-ish frame and box look. Zulip comes with all the usual features, such as chat forums and private messaging. The bot and the plugins are where things start to heat up. Users with appropriate authorizations can add one or more bots to their profile that either send outgoing messages to webhooks or provide a webhook for incoming messages themselves.

Zulip supports a number of popular JSON formats from other services such as GitHub, GitLab, or Ansible. For example, users can create a Gitea webhook URL for one of their bots and then enter the URL in their Gitea repository. When events occur in the repository (e.g., commit or pull requests), Gitea then sends a message to the Zulip forum or directly to the user (**Figure 1**).

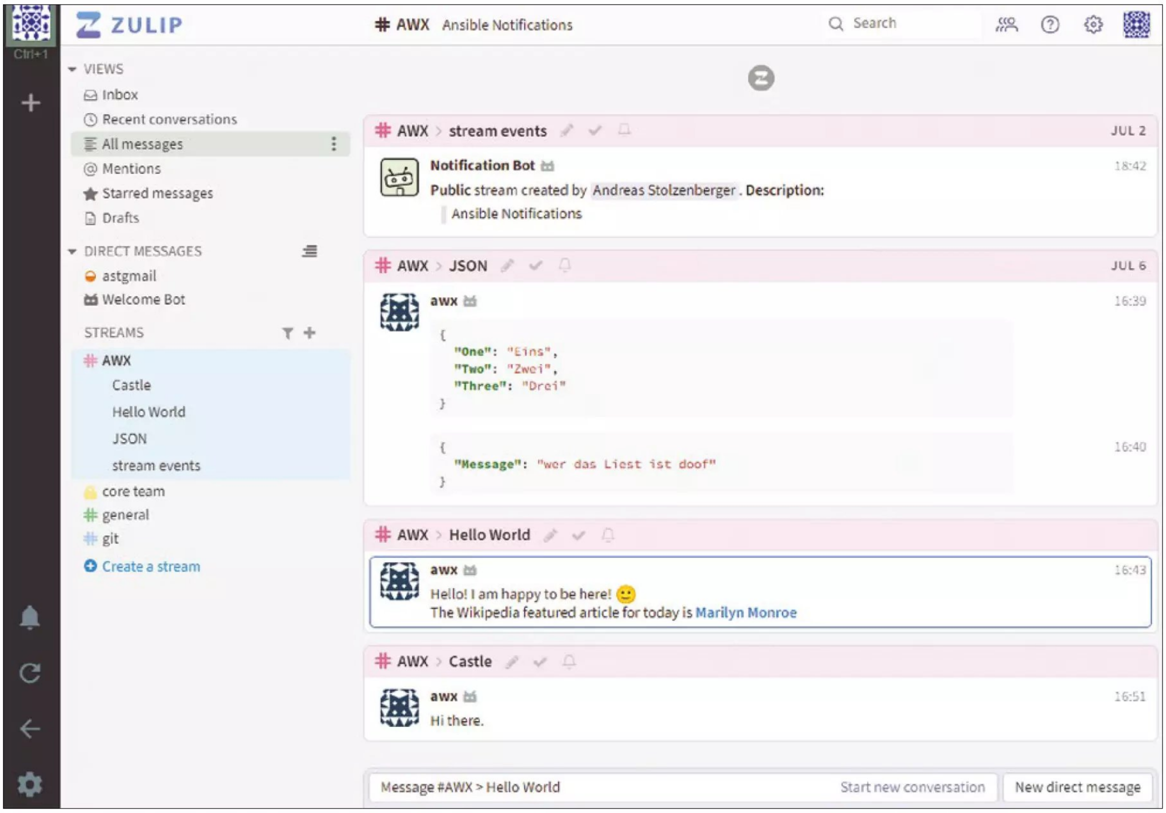
The bot's account can also be used for external access by REST API. Outgoing bots transmit messages from the active forums to webhook URLs. Users can specify whether to use the Slack or Zulip format. If external tools have an incoming webhook for Slack, they will also work with Zulip. If you need more than the simple webhook and REST API functionality offered by the Zulip bots, you can also program your own bots in Python. The Zulip documentation describes how this works in detail.

Zulip provides a small management tool for housekeeping, but if the server is containerized, the tool requires some overhead. You can use it to reset passwords or create users without detouring with an invite mail. Theoretically, `manage.py` could also create a backup of your Zulip environment directly, but this only works correctly in a traditional VM setup with Postgres and Zulip running on the same system. In a containerized setup, the backup tool does not have access to the database server because it runs in a different container. The `manage.py backup` command will only work if you specify the `--skip-db` option, and you will also need to back up the database in the PostgreSQL container yourself.

The Zulip server is fast and easy to set up with Podman. The flexible bot system is ideal for anyone who wants to integrate the chat platform with other tools. The free open source version is fully featured.

Mattermost

Mattermost was originally created as a proprietary, in-house communication tool for a game development company. It was outsourced as an open source project in 2015 and has been maintained by Mattermost Inc. ever since; in fact, the company was



AWX > stream events JUL 2

Notification Bot 18:42
Public stream created by Andreas Stolzenberger. Description:
Ansible Notifications

AWX > JSON JUL 6

awx 16:39

```
{
  "One": "Eins",
  "Two": "Zwei",
  "Three": "Drei"
}
```

16:40

```
{
  "Message": "wer das Liest ist doof"
}
```

AWX > Hello World

awx 16:43
Hello! I am happy to be here! 🙂
The Wikipedia featured article for today is Marilyn Monroe

AWX > Castle

awx 16:51
Hi there.

Message #AWX > Hello World

Figure 1: Incoming bots evaluate incoming messages with previously configured JSON parsers. The JSON formatter displays the raw format and helps you build your own bots.

founded specifically for this purpose and offers commercial support contracts for Mattermost installations and additional features that are not included in the open source version.

Mattermost only needs a PostgreSQL database and its own server for operation. The server could take care of SSL but prefers to hide behind an SSL-terminating reverse proxy; therefore, you can run Mattermost as a containerized solution on Docker, Podman, or Kubernetes. I used a classic VM in this example.

The installation on CentOS Stream 9 is very simple, involving just two steps – database preparation [6] and service setup [7] – that take just a few minutes to complete. The Enterprise Linux (EL) setup uses a tarball image, which makes upgrading the server to new versions slightly more complex. The setup on Debian and Ubuntu is easier to manage, because a PPA repository for these platforms is available, and installation and upgrades are therefore easy with `apt`.

For a straightforward function test on the LAN, I did not use an HTTPS setup with proxy. HTTP access to port 8065 on the server was fine. As with Zulip, you only need a web browser to access the chat server. Clients for various desktop and cellphone platforms are also available. The Mattermost UI looks a little tidier and more modern than Zulip, although users do have to search through submenus for some functions, which are probably easier to find in the Zulip UI.

The system console allows you to manage the server and its functions, as well as teams and users. The `mmctl` admin command-line interface (CLI) tool is on the server itself. The premium functions that are only available to commercial users can also be found in the system console. They primarily include tools for integrating the Mattermost server with an Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) user directory and the single sign-on services. Guest access with restricted rights for users from outside the organization is also unavailable in the open source version.

Independent Bots

Like Zulip, Mattermost supports bots for incoming or outgoing webhooks. In Mattermost, though, the bots do not run in the scope of a user account, but as independent users. Unfortunately, the incoming hooks in particular lack flexibility. The formatting of the incoming JSON array is quite rigid and only recognizes the Slack format. Zulip already supports a variety of messages with different formatting and offers the user an option to write customized bots for JSON parsing in Python.

Mattermost adds further functionality with apps that you can install from an app store or upload to its server as plugins (Figure 2). These apps provide additional chat features such as polls or improved integration with external services such as GitHub. Users can write their own plugins and integrate them into the solution. Besides group chats, Mattermost comes with a playbook function, and workflows for teams can be modeled and started here. Mattermost then informs the users involved when they have tasks to complete in the scope of the playbook. Again, users of the free version only get part of the functionality. Playbook steps such as a due date in a task assignment are only available for commercial customers.

The software does not come with an integrated backup script, although the documentation describes precisely which files and directories you need to back up along with the PostgreSQL database dump. Mattermost has a slightly more modern UI and a simpler back end than Zulip, tends to rely more on apps and playbook workflow functions, is somewhat less flexible in terms of bot and webhook design, and does not give users of the free version access to the full feature set.

Rocket.Chat

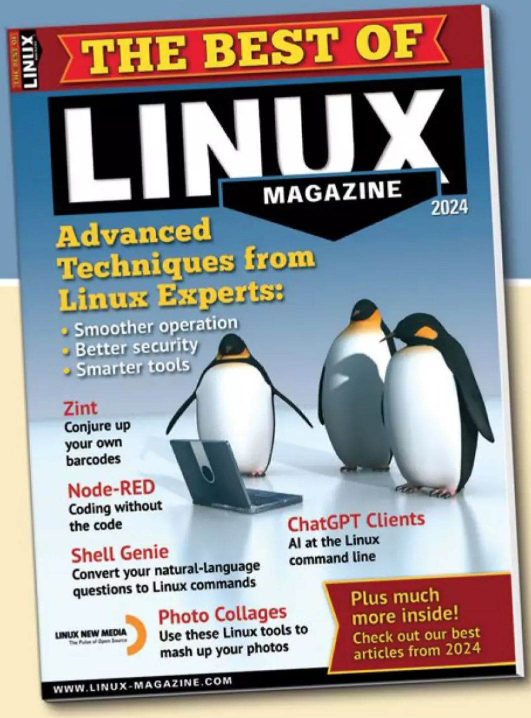
The Rocket.Chat server can run as a Snap or in a containerized setup. In the lab, I used Docker on a CentOS Streams 9 system and relied on the included Docker Compose file to start the service. Rocket.Chat needs two containers: one with MongoDB for data storage and another for its own server. The server shares its web interface on HTTP port 3000, so you can set up your own SSL-terminating proxy there.

At the start of the install, the self-hosted server creates a system administrator and then prompts you to register with Rocket.Chat. Of course, this is not in your interest for a simple test installation on the local LAN or an isolated network. Unfortunately, the tool does not have a simple button if you want to skip this process or are not interested in online registration. A lengthy search in the documentation revealed a reference to a configuration option in the small print that says to add

```
OVERWRITE_SETTING_Show_Setup_Wizard=↩
  completed
```

to the `Rocker.Chat.Extra.env` configuration file. In this example with Docker Compose, though, you need to add the following option to the Compose file:

```
environment:
...
  OVERWRITE_SETTING_Show_Setup_Wizard: ↩
    completed
```


THE BEST OF LINUX MAGAZINE 2024
Advanced Techniques from Linux Experts:
• Smoother operation
• Better security
• Smarter tools
Zint Conjure up your own barcodes
Node-RED Coding without the code
Shell Genie Convert your natural-language questions to Linux commands
ChatGPT Clients AI at the Linux command line
Photo Collages Use these Linux tools to mash up your photos
Plus much more inside! Check out our best articles from 2024
WWW.LINUX-MAGAZINE.COM

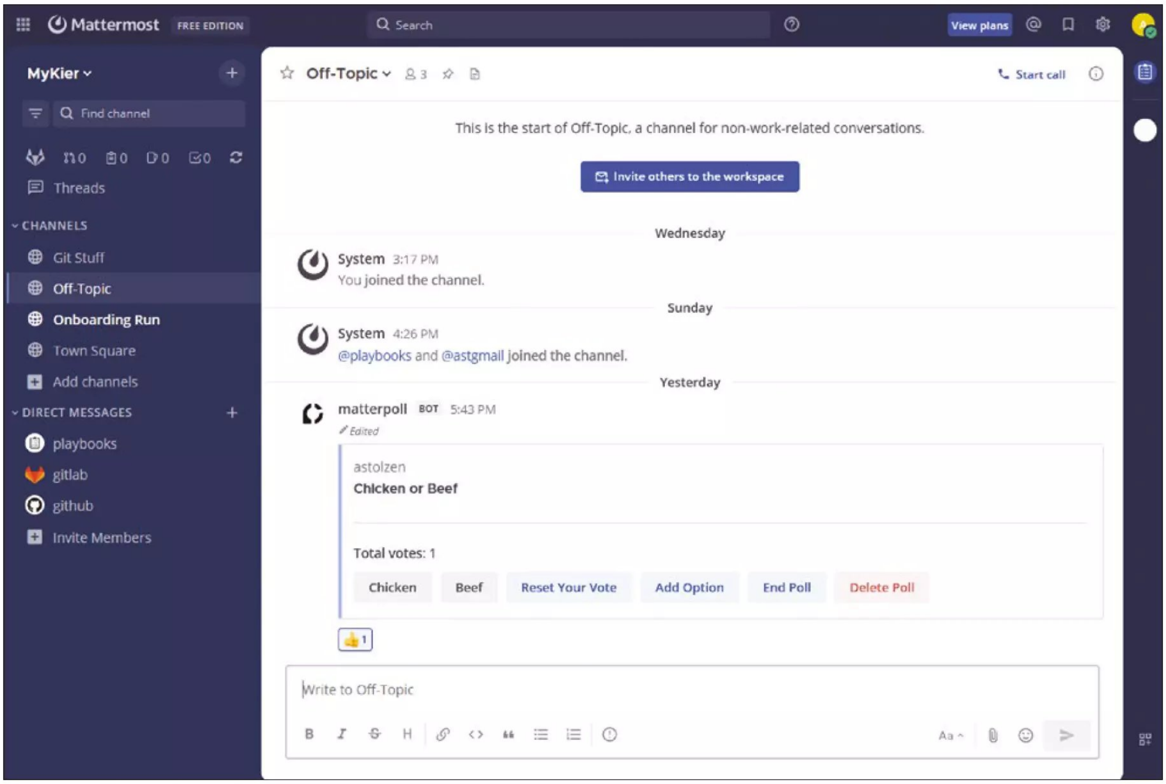
THE BEST OF LINUX MAGAZINE 2024

Advanced Techniques from Linux Experts

This new special edition brings you the best and most practical articles from the 2024 editions of *Linux Magazine*.

Whether you're new to Linux or a seasoned veteran, you'll find something useful in the tips, tools, and technologies inside.

ORDER ONLINE:
shop.linuxnewmedia.com/shop


Mattermost
FREE EDITION
Search
View plans
MyKier
Find channel
Threads
CHANNELS
Git Stuff
Off-Topic
Onboarding Run
Town Square
Add channels
DIRECT MESSAGES
playbooks
gitlab
github
Invite Members
Off-Topic
Start call
Invite others to the workspace
Wednesday
Sunday
Yesterday
Chicken
Beef
Reset Your Vote
Add Option
End Poll
Delete Poll
Write to Off-Topic

This is the start of Off-Topic, a channel for non-work-related conversations.

System 3:17 PM
You joined the channel.

System 4:26 PM
@playbooks and @astgmail joined the channel.

matterpoll BOT 5:43 PM
Edited

astolzen
Chicken or Beef

Total votes: 1

Figure 2: **Mattermost uses slash commands to launch active plugins.**

As with other tools, Rocket.Chat can be run in your browser, on your cellphone, or as a desktop app. The web UI is very much modeled on market leader Slack (Figure 3). Rocket.Chat's main focus is on the chat functionality and integrating chat with other applications.

In every second menu, the application makes users of the community edition feel that they would be far better off with a licensed version rather than the free edition. All of the advanced tools (e.g., AD/Security Assertion Markup Language (SAML)/LDAP integration, the app marketplace, device management, the dashboard) are only available to users of the commercial version.

Develop Your Own Applications

That said, users do have the option of extending their setup even without ready-made applications from the Rocket.Chat Marketplace. The manufacturer offers the Apps-Engine CLI toolkit for developing your own Rocket.Chat applications. Armed with this toolkit and TypeScript skills to match, you can write your own plugins and add them to your chat servers. Unfortunately, there is a catch: The Apps-Engine CLI cannot be installed in an environment set up with Docker Compose, which means you need a second, legacy environment on a VM for application development. This Rocket.Chat setup on an Ubuntu 24.04 VM by the Snap package manager was a smooth experience. Strangely, Snap does not use the typical Linux server directory structure but drops the complete installation, including MongoDB, in directories somewhere under `/var/snap`, where it also stores the configuration files. The Snap setup also configures the MongoDB server without security measures (e.g., access passwords).

After installing Node.js and the required dependencies, the `rc-apps` command creates the desired development environments for Rocket.Chat extensions. You can roll out ready-to-use applications on the platform by typing

```
rc-apps --deploy
```

and then assign slash commands to match. Applications follow the usual

```
/myapp parameter1 parameter2 [...]
```

pattern for launching from within the chat. When it comes to webhook-based bot integration, Rocket.Chat is on a par with Mattermost. You have simple hooks, as well, but only a few options for customizing JSON parsing for the bot payload.

Conclusions

All three services presented here offer a simple approach to setting up self-hosted chat platforms for corporate communication. Even though Zulip might not have the prettiest UI, it provides the widest range of functions in the open source version. For one thing, it doesn't hide any features behind a paywall, and for another, it offers simple and open customization options for bots, API access, and customized webhooks.

The strength of Mattermost is playbooks. If used correctly, they make it very easy to model and implement complex workflows in a chat tool. It is a pity, though, that the provider selectively removed individual modules, or even selected functions such as the playbook appointment option, from the open source version.

Rocket.Chat, which sees itself primarily as a commercial option for Slack users, definitely comes with the biggest limitations in the open source version. In terms of functionality, the platform tries to keep up with the scope and look of its role model Slack, which it does well. However, the free version feels too limited in terms of the feature set and more like a demo version than a community solution.

In terms of installation variants, a containerized setup with Docker or Podman offers the greatest convenience. However, you do need to check very carefully which images are integrated into your environments. Containerized operation also makes it more difficult to use CLI tools, if the platform offers them. The classic setup on a VM with your package manager or a tarball offers more granular control over the configuration, version, and security of the individual components. A containerized setup in which the operator only allows tested and verified components would be ideal. ■

Info

[1] Zulip: [https://github.com/zulip/zulip]

[2] Mattermost: [https://github.com/mattermost/mattermost]

[3] Rocket.Chat: [https://www.rocket.chat]

[4] "Manage Containerized Setups with Ansible" by Andreas Stolzenberger, *ADMIN*, issue 77, 2023, pg. 64, [https://www.admin-magazine.com/Archive/2023/77/Manage-containerized-setups-with-Ansible/]

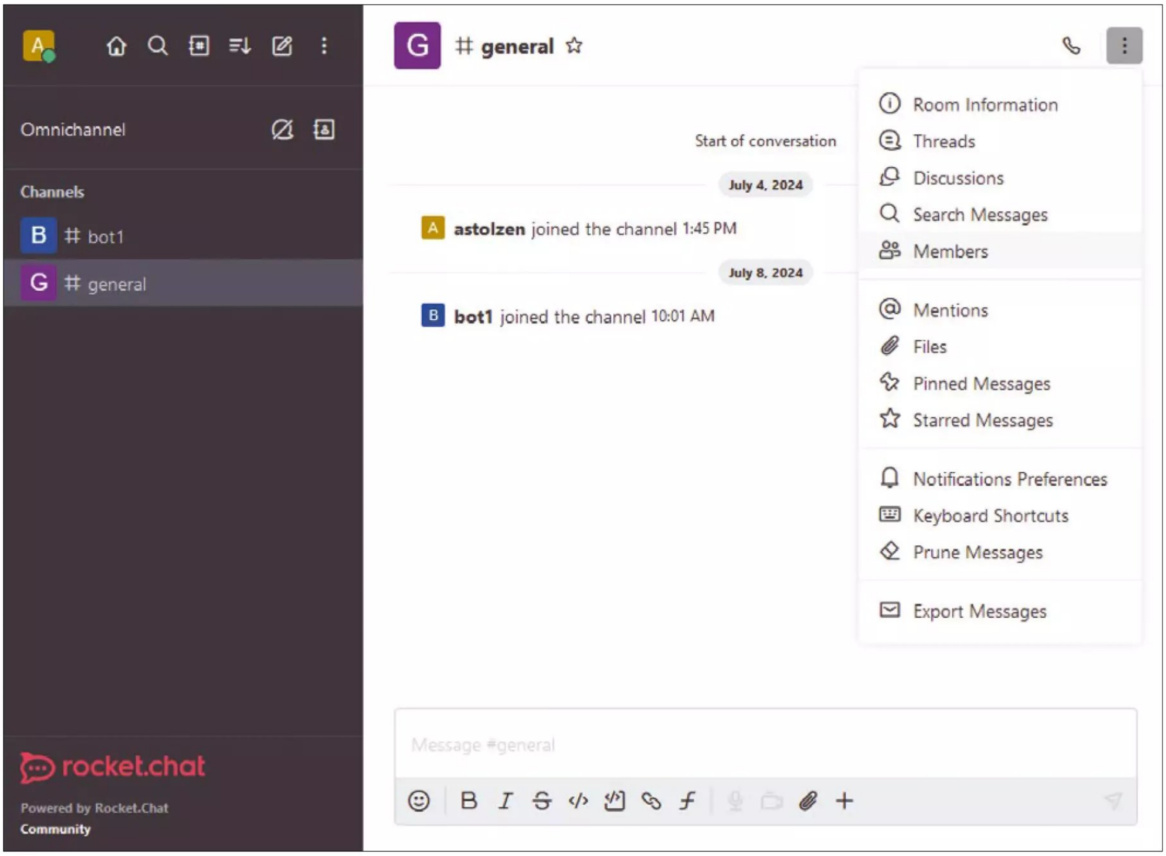
[5] Docker Compose file: [https://github.com/zulip/docker-zulip/blob/main/docker-compose.yml]

[6] Database preparation for Mattermost: [https://docs.mattermost.com/install/prepare-mattermost-database.html]

[7] Service setup for Mattermost: [https://docs.mattermost.com/install/install-rhel.html]

The Author

Andreas Stolzenberger worked as an IT magazine editor for 17 years. He was the deputy editor in chief of the German *Network Computing* magazine from 2000 to 2010. After that, he worked as a solution engineer at Dell and VMware. In 2012 Andreas moved to Red Hat. There, he currently works as principal solution architect in the Technical Partner Development department.



A
Omnichannel
Channels
B # bot1
G # general
rocket.chat
Powered by Rocket.Chat
Community
G # general
Room Information
Threads
Discussions
Search Messages
Members
Mentions
Files
Pinned Messages
Starred Messages
Notifications Preferences
Keyboard Shortcuts
Prune Messages
Export Messages
Message #general

Start of conversation

July 4, 2024

A **astolzen** joined the channel 1:45 PM

July 8, 2024

B **bot1** joined the channel 10:01 AM

Figure 3: The Rocket.Chat UI is based on the Slack model.

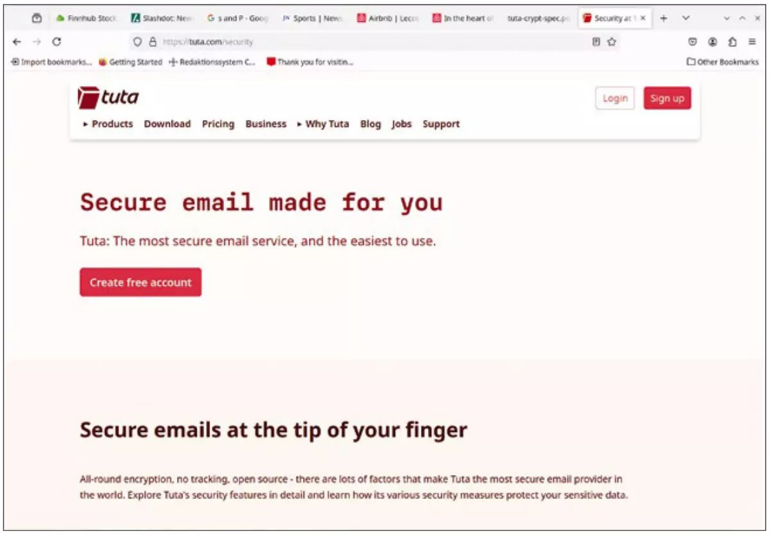
unsigned int len = cp_count * sizeof(*grouplist);
if (copy_from_user(group_info->blocks[i], grouplist, len))
return -EFAULT;
grouplist += NGROUPS_PER_BLOCK;
count -= cp_count;
return 0;
struct group_info *groups_alloc(int gidsetsize)
struct group_info *group_info;
int nblocks;
int i;
using System.Collections.Generic;
using System.Linq;
using System.Text.RegularExpressions;
namespace Rextester
public class Program
public static void Main(string[] args)
//Your code goes here
Console.WriteLine("Hello, world!");

Post-Quantum Cryptography with TutaCrypt

Future Proof

Your encrypted email messages might not be so secret after the quantum revolution picks up steam. Tuta Mail comes with a quantum-resistant alternative for users who take the long view. By Valentin Franck

Quantum computing represents a seismic shift in technology, promising breakthroughs in numerous fields. But for cryptography, quantum computers are a looming threat. Experts are well aware that traditional asymmetric encryption methods are at risk from quantum algorithms such as Shor's algorithm [1], which efficiently factorizes large numbers and could theoretically break RSA and elliptic curve cryptography (ECC). For symmetric encryption techniques such as AES, the quantum algorithm known as Grover's algorithm [2] halves the effective key length, thus requiring longer keys to maintain security. The looming end of conventional cryptography has caused mathematicians to search for a new kind of encryption that will resist quantum attacks. In order to succeed in the post-quantum world, a new generation of cryptographic algorithms will need to avoid the dependence on the discrete logarithm problem and the hardness of integer factorization of semiprimes characterizing classical approaches and rely, instead, on hashes, lattices, and other advanced techniques. Although quantum computers are still at the experimental stage, the IT world is already getting restless to know what will come next. Most cryptography customers want to encrypt their data permanently, not just until a bigger computer arrives. For that reason, some users are asking if it is possible to build quantum protection into existing crypto products now.


Finnhub Stock
Slashdot: New
s and P - Goo
Sports | New
Airbnb | Lecc
In the heart o
tuta-crypt-spec.p
Security at T
https://tuta.com/security
Import bookmarks...
Getting Started
Redaktionssystem C...
Thank you for visitin...
Other Bookmarks
tuta
Login
Sign up
▸ Products
Download
Pricing
Business
▸ Why Tuta
Blog
Jobs
Support
Create free account

Secure email made for you

Tuta: The most secure email service, and the easiest to use.

Secure emails at the tip of your finger

All-round encryption, no tracking, open source - there are lots of factors that make Tuta the most secure email provider in the world. Explore Tuta's security features in detail and learn how its various security measures protect your sensitive data.

Figure 1: Post-quantum encryption is a logical development for Tuta Mail, which has always been security focused.

Lead Image © pitinan, 123RF.com

With that need in mind, the email vendor Tuta [3] just rolled out a new version of their Tuta Mail product (Figure 1) that features quantum-resistant encryption, becoming one of the first commercial vendors to tackle this perplexing problem.

The Algorithms

TutaCrypt [4] uses a combination of relatively new quantum-resistant algorithms and well-proven classical encryption to offer a hybrid protocol. The quantum-resistant component is Kyber [5], which was standardized as ML-KEM. Kyber, which is an outcome of NIST's Post-Quantum Cryptography competition, is a lattice-based key encapsulation mechanism (KEM). Its security relies on the Module Learning With Errors (M-LWE) problem, a difficult mathematical challenge even for quantum computers. In addition to its security properties, Kyber-1024 Key Pair offers excellent computational efficiency and comparatively small key and ciphertext sizes, making it well suited for resource-constrained environments such as mobile devices.

TutaCrypt also uses two Diffie-Hellman key exchanges over Curve25519 (X25519) between static long-term identity keys and an ephemeral sender key (for randomized encryption and partial forward secrecy). X25519 is a well tested technique for Elliptic Curve Diffie-Hellman (ECDH) key exchange that provides strong security against classical adversaries. At the same time, the X25519 approach is highly efficient, requiring very short key sizes and computation times. In addition to safeguarding against impersonation attacks, X25519 ensures confidentiality against non-quantum adversaries even if weaknesses are discovered in Kyber in the future.

Once the shared secrets are established using Kyber and X25519, they are combined to derive a key for symmetric encryption with HKDF. TutaCrypt splits the key and encrypts email data with AES256 in CBC mode and then computes a tag over the resulting ciphertext with HMAC-SHA256, which yields authenticated encryption. AES is quantum-resistant with sufficiently long keys (256 bits) and provides efficient, authenticated encryption for bulk data. This hybrid approach is both tamperproof and highly efficient. By using both classical and post-quantum algorithms (Figure 2), TutaCrypt protects against unknown vulnerabilities in both realms. And the quantum-resistant components ensure that encrypted data intercepted today will remain secure against attacks from a future quantum computer.

The secret keys are wrapped with a password-derived key (Argon2id) and stored on Tuta's own servers in data centers in Germany. This approach means only the user can decrypt their private key to securely access their encrypted mailboxes, calendars, and contacts. TutaCrypt encrypts the entire email, including the body, subject line, and attachments, using this hybrid protocol. Using Elliptic Curve Diffie-Hellman (X25519), two shared secrets are exchanged between

- the sender's long-term identity key (IKA) and the recipient's identity key (IKB), and
- the sender's ephemeral key (EKA) and the recipient's identity key (IKB).

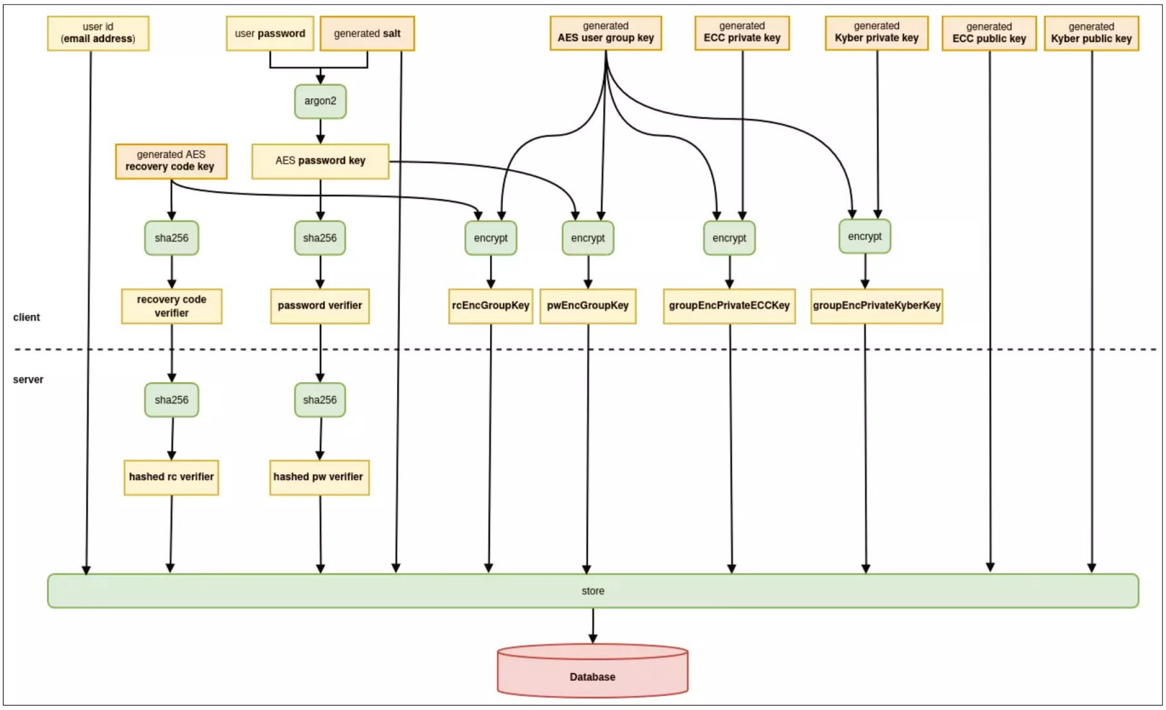

user id (email address)
user password
generated salt
generated AES user group key
generated ECC private key
generated Kyber private key
generated ECC public key
generated Kyber public key
argon2
generated AES recovery code key
AES password key
sha256
sha256
encrypt
encrypt
encrypt
encrypt
recovery code verifier
password verifier
rcEncGroupKey
pwEncGroupKey
groupEncPrivateECCKey
groupEncPrivateKyberKey
client
server
sha256
sha256
hashed rc verifier
hashed pw verifier
store
Database

Figure 2: Tuta's hybrid algorithm includes both classical and post-quantum components.

TutaCrypt for Organizations

Tuta Mail's desktop, mobile, and web clients all come with the hybrid, quantum-safe encryption protocol. The desktop client is currently built for the following platforms:

- Windows as an nsis-installer
- macOs as a DMG for install and a ZIP for updates
- Linux as a self-contained AppImage executable

On Windows, you can set up the desktop client in enterprise mode by passing the following flags to the installer:

- `/S` – install silently (no wizard)
- `/allusers` – install for all users on the system rather than just the current user (requires privilege escalation)
- `/disableAutoUpdates` – disable automatic updates (updates would need to be done manually by an admin)

You can uninstall silently by passing `/S` to the uninstaller (found in the installation directory under *Program Files*). Administrators can bulk import employee email addresses using a CSV file with the following data:

```
*First Name,Last Name,➘
Email Address,Role,➘
Temporary Password*
```

After uploading the file via Tuta's admin console, users will need to receive credentials (email address and password, possibly also a second factor) and a link to download the Tuta Mail client. The Tuta admin settings allow the admin to manage users, change passwords, and revoke access in the case of a security incident.

These secrets are referred to as DHI and DHE, respectively. A third shared secret, SSPQ, is exchanged using Kyber-1024's key encapsulation mechanism. The three shared secrets (DHI, DHE, SSPQ) are then combined via an HKDF-SHA-256 function to derive the encryption keys, and the keys are then used to encrypt and decrypt the message, ensuring confidentiality, integrity, and authenticity without requiring signatures.

TutaCrypt is already the default for newly created Tuta Mail accounts. Existing accounts still using traditional AES/RSA algorithms for asymmetric encryption will gradually transition to quantum-safe encryption once the key rotation mechanism is fully implemented. This phased rollout ensures minimal disruption while offering enhanced security for all Tuta users as quickly as possible. Tuta Mail is committed to ongoing improvements in cryptography. The company is currently working with partners at the University of Wuppertal to have TutaCrypt formally reviewed and verified. Tuta is also at work on new key verification mechanisms that will provide cryptographically guaranteed authentication. See the box entitled "TutaCrypt for Organizations" for more on how to implement Tuta's quantum-resistant cryptography in institutional settings.

Conclusion

Cryptography experts are already writing about all the ways quantum computers will break conventional encryption algorithms, and you can bet that spies and cybercriminals are well aware of the potential opportunities this will create. Given the rapid pace of development in the IT space, the "harvest now, decrypt later" (HNDL) attack is now a significant concern. Attackers could intercept encrypted communications today and store them until quantum computers are capable of decrypting them – and no one can predict when this is going to happen – next year or 10 years from now. TutaCrypt mitigates this risk by using quantum-resistant algorithms, ensuring data remains secure now as well as in the future. ■

Info

[1] Shor's Algorithm: [https://en.wikipedia.org/wiki/Shor%27s_algorithm]

[2] Grover's Algorithm: [https://en.wikipedia.org/wiki/Grover%27s_algorithm]

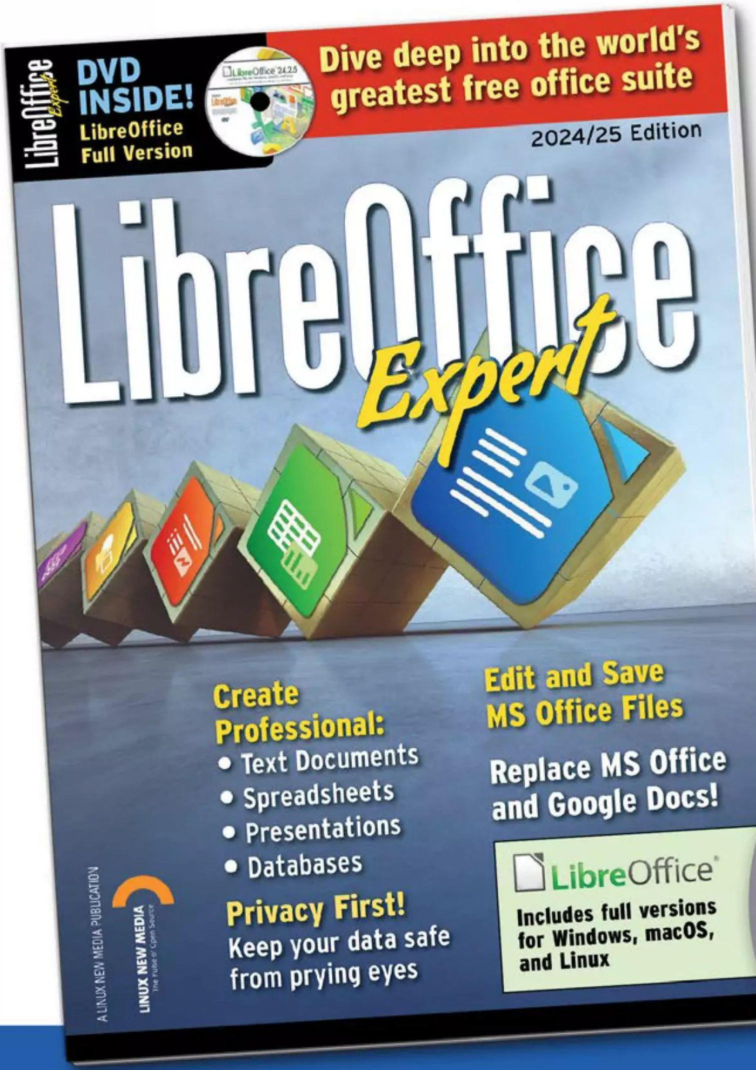
[3] Tuta: [https://tuta.com/]

[4] TutaCrypt: [https://tuta.com/documents/tuta-crypt-spec.pdf]

[5] Kyber: [https://pq-crystals.org/kyber/index.shtml]

Shop the Shop
shop.linuxnewmedia.com/shop

Become a LibreOffice Expert



LibreOffice Expert

DVD INSIDE!
LibreOffice Full Version

Dive deep into the world's greatest free office suite

2024/25 Edition

LibreOffice Expert

Create Professional:
- Text Documents
- Spreadsheets
- Presentations
- Databases

Privacy First!
Keep your data safe from prying eyes

Edit and Save MS Office Files

Replace MS Office and Google Docs!

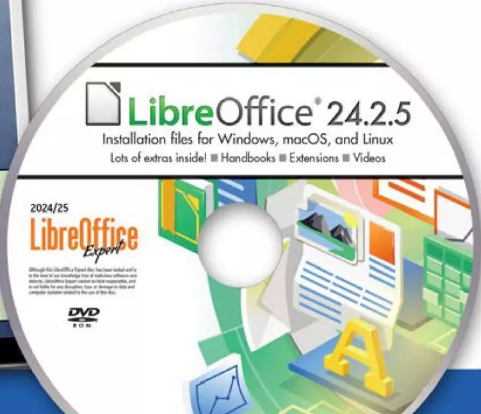
LibreOffice
Includes full versions for Windows, macOS, and Linux

A LINUX NEW MEDIA PUBLICATION
LINUX NEW MEDIA
The Pulse of Open Source

Explore the FREE office suite used by busy professionals around the world!

Create Professional:
- Text Documents
- Spreadsheets
- Presentations
- Databases

Whether you work on a Windows PC, a Mac, or a Linux system, you have all you need to get started with LibreOffice today. This single-volume special edition will serve as your guide!



LibreOffice® 24.2.5
Installation files for Windows, macOS, and Linux
Lots of extras inside! ■ Handbooks ■ Extensions ■ Videos

2024/25
LibreOffice Expert

DVD ROM

Order online:
shop.linuxnewmedia.com/shop

For Windows, macOS, and Linux users!

Serverless applications with OpenFaaS

Doing Without

Use advanced techniques for automation, optimization, and security to deploy serverless applications with the OpenFaaS framework in a Linux environment. By Marcin Gastol

In the rapidly evolving world of software development, one of the most transformative shifts in recent years has been the rise of serverless computing. This approach allows developers to focus on building and deploying code without having to worry about the underlying infrastructure. OpenFaaS (function as a service) [1] is a popular open source framework that enables serverless functions on any infrastructure, including public cloud environments, private data centers, or even on-premises Linux servers.

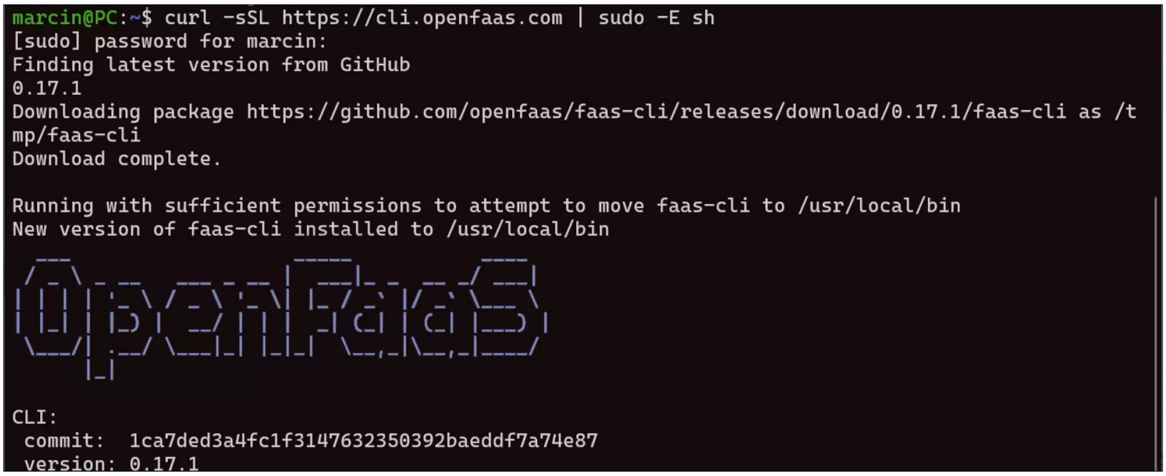
Environment Setup

Before diving in, ensure you have the following prerequisites in place:

- Ubuntu Server: an on-premises machine or cloud virtual machine (VM) running Ubuntu (20.04 LTS or later is recommended) with sudo privileges.
- Kubernetes cluster: OpenFaaS runs on Kubernetes, so you need access to a Kubernetes cluster.
- Docker: to build and push container images for your serverless functions. Install Docker on your Ubuntu machine (if using MicroK8s, Docker is optional because it uses containerd, but having Docker is useful for the OpenFaaS command-line interface (CLI)).
- OpenFaaS CLI: `faas-cli` for managing functions. The quickest way to install it on your machine is with the OpenFaaS-provided script (Figure 1):

```
curl -sSL https://cli.openfaas.com | ➘
  sudo sh
```

With these prerequisites satisfied, your Ubuntu system should have a running Kubernetes cluster, `kubectl` configured to communicate with your cluster, and Docker and `faas-cli`. Now you can deploy your first serverless function and scale and manage it. OpenFaaS supports deploying functions through the web user interface (UI), CLI, or GitOps. In this example, I use the CLI for precision and repeatability.



```
marcin@PC:~$ curl -sSL https://cli.openfaas.com | sudo -E sh
[sudo] password for marcin:
Finding latest version from GitHub
0.17.1
Downloading package https://github.com/openfaas/faas-cli/releases/download/0.17.1/faas-cli as /t
mp/faas-cli
Download complete.

Running with sufficient permissions to attempt to move faas-cli to /usr/local/bin
New version of faas-cli installed to /usr/local/bin
  ___                   _____           ____
 / _ \ _ __   ___ _ __ |  ___|_ _  __ _/ ___|
| | | | '_ \ / _ \ '_ \| |_ / _` |/ _` \___ \
| |_| | |_) |  __/ | | |  _| (_| | (_| |___) |
 \___/| .__/ \___|_| |_|_|  \__,_|\__,_|____/
      |_|


CLI:
 commit:  1ca7ded3a4fc1f3147632350392baeddf7a74e87
 version: 0.17.1
```

Figure 1: Installing OpenFaaS.

Photo by Kier in Sight Archives on Unsplash

Deployment

In a real-world scenario, you'll write your own functions. With OpenFaaS, you can use the `faas-cli new` command to scaffold a new function in your language of choice (templates for Node.js, Python, Go, etc., are available). I won't go deep into function development in this tutorial, but as an example, consider

```
faas-cli new --lang python my-function
```

which creates `stack.yaml` (deployment manifest) and a `my-function` folder with a handler, `handler.py` (Figure 2). You can implement your logic there, then build and deploy with

```
faas-cli up -f stack.yaml
```

This command builds a Docker image, pushes it, and deploys the function in one step (Figure 3). Ensure you're logged in to a container registry or use the OpenFaaS default (pushes to Docker Hub, or, adjust `image:` in the YAML file).

For now, stick with the example function to focus on deployment mechanics.

Function Scaling

One of the powerful features of OpenFaaS is automatic horizontal scaling of functions in response to demand. When a function is deployed, by default it runs with one replica (pod). The OpenFaaS autoscaler monitors incoming requests and scales the number of function replicas between a minimum and maximum value.

In OpenFaaS Community Edition (CE) [2], autoscaling is handled by Prometheus and AlertManager. The gateway and function pods expose metrics (e.g., request rates). AlertManager is configured with rules that trigger when the requests per second for a function exceed a threshold. When triggered, AlertManager sends an alert to the OpenFaaS gateway's `/system/alert` endpoint, which in turn scales up the function's Deployment (increasing replicas). Similarly, if load drops off, alerts can scale the function down (but not below the configured minimum). This process happens behind the scenes automatically.

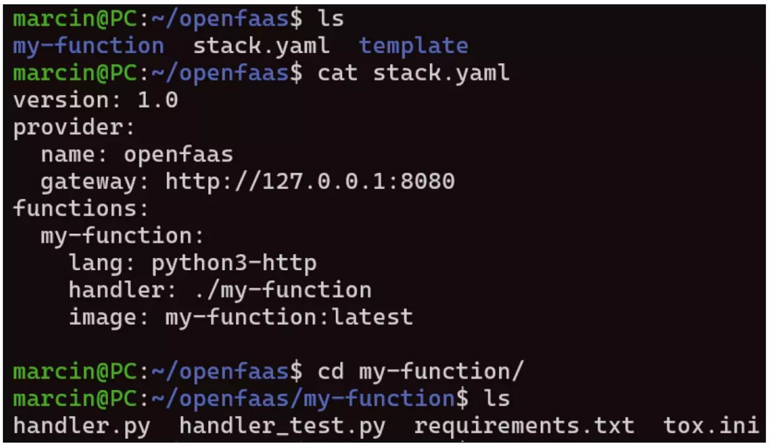
Scale to Zero

OpenFaaS can scale functions down to zero replicas when idle with the `faas-idler` component to save resources and scale them back up on demand. In OpenFaaS Pro, you simply mark functions with `com.openfaas.scale.zero=true` to opt in to this behavior. In the Community Edition, scale to zero can be enabled by deploying `faas-idler` and labeling functions accordingly. When enabled, an idle function (no requests for a configured duration, e.g., five minutes) will be scaled down to zero pods, and the next request will cold-start the function by scaling it back up. Cold starts add a bit of latency, so use this feature judiciously for non-latency-sensitive workloads.

Manual Scaling

A function can also be scaled manually, if needed (e.g., pre-warm more replicas in anticipation of a load spike). Because functions are just Kubernetes Deployments, you can use `kubectl` or `faas-cli` to scale:

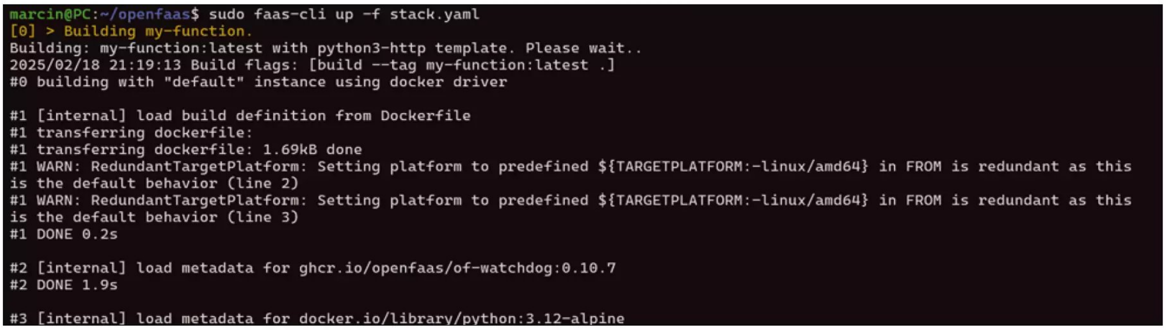
```
kubectl scale deploy ↵
  -n openfaas-fn figlet --replicas=3
```



```
marcin@PC:~/openfaas$ ls
my-function  stack.yaml  template
marcin@PC:~/openfaas$ cat stack.yaml
version: 1.0
provider:
  name: openfaas
  gateway: http://127.0.0.1:8080
functions:
  my-function:
    lang: python3-http
    handler: ./my-function
    image: my-function:latest

marcin@PC:~/openfaas$ cd my-function/
marcin@PC:~/openfaas/my-function$ ls
handler.py  handler_test.py  requirements.txt  tox.ini
```

Figure 2: Folder structure and file content created by a `faas-cli new` command.



```
marcin@PC:~/openfaas$ sudo faas-cli up -f stack.yaml
[0] > Building my-function.
Building: my-function:latest with python3-http template. Please wait..
2025/02/18 21:19:13 Build flags: [build --tag my-function:latest .]
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile:
#1 transferring dockerfile: 1.69kB done
#1 WARN: RedundantTargetPlatform: Setting platform to predefined ${TARGETPLATFORM:-linux/amd64} in FROM is redundant as this
is the default behavior (line 2)
#1 WARN: RedundantTargetPlatform: Setting platform to predefined ${TARGETPLATFORM:-linux/amd64} in FROM is redundant as this
is the default behavior (line 3)
#1 DONE 0.2s

#2 [internal] load metadata for ghcr.io/openfaas/of-watchdog:0.10.7
#2 DONE 1.9s

#3 [internal] load metadata for docker.io/library/python:3.12-alpine
```

Figure 3: The `faas-cli up` command builds, pushes, and deploys.

This command sets three replicas for the `figlet` function. The OpenFaaS gateway will respect this parameter (i.e., it won't scale down below three on its own if you've set the minimum with this manual action because it updates the Deployment). The command

```
faas-cli scale figlet --replicas=3
```

demonstrates the CLI `scale` command.

Security Best Practices

Running serverless functions on your infrastructure introduces security considerations at the platform level (OpenFaaS and Kubernetes) and the function level (your code). OpenFaaS, being Kubernetes-native, inherits many of the Kubernetes security features and adds its own safeguards. The following are key security best practices to consider for an OpenFaaS deployment, including access control and network restrictions.

Gateway Authentication and Access Control

As discussed, OpenFaaS gateway comes with basic authentication enabled by default. Always keep authentication enabled on the gateway, especially if the gateway is exposed on a public network. Use a strong, randomly generated admin password and rotate it, if needed. (You can update the Kubernetes secret and restart the gateway.)

If you have multiple users or teams deploying functions, consider leveraging Kubernetes role-based access control (RBAC). You can control access to the OpenFaaS Kubernetes resources (e.g., Function custom resource definitions (CRDs) or Deployments) via Kubernetes roles. The OpenFaaS Pro operator uses a Function CRD with RBAC, meaning users could be granted permission to create or update functions in specific namespaces, rather than sharing the admin gateway credentials. In the Community Edition, you might achieve something similar by controlling who can call the OpenFaaS REST API or `faas-cli` (perhaps by not exposing the gateway externally and only allowing anyone through a proxy or API gateway under your control).

Traffic Encryption (TLS)

Enable TLS encryption for any client traffic to the OpenFaaS gateway. If you deployed on Azure Kubernetes Service (AKS) with a LoadBalancer service type, the gateway endpoint is HTTP by default. The simplest way is with an Ingress Controller (e.g., NGINX or Traefik) with cert-manager to automatically fetch Let's Encrypt certificates. The OpenFaaS docs [3] recommend using an Ingress with TLS termination for production. For example, you might deploy NGINX Ingress, configure a host such as `faas.example.com` for the gateway, and let cert-manager generate a certificate for it. Alternatively, the OpenFaaS gateway can be configured with TLS directly, but managing certificates by Kubernetes Ingress is more common.

Network Restrictions and Isolation

By default, any pod in Kubernetes can talk to any other pod (unless restricted by NetworkPolicies). In an OpenFaaS context, you might want to limit network access in a few ways:

- Namespace Isolation: If you have multiple function namespaces (for different environments or tenants), consider applying Kubernetes NetworkPolicies to prevent functions in one namespace from calling functions in another or from directly calling the OpenFaaS gateway in unauthorized ways. For example, you can allow traffic from the gateway to functions, but disallow cross-namespace function-to-function traffic.
- Restrict External Access: If certain functions should not access the Internet (for compliance or safety), use NetworkPolicy to block egress from those function pods to external IPs, except perhaps to specific services.
- API Server Access: By default, OpenFaaS functions are not given permissions to talk to the Kubernetes API. The OpenFaaS RBAC rules ensure functions run with a limited service account that cannot read or write Kubernetes resources. This good security measure prevents a compromised function from manipulating cluster infrastructure. Only grant a function access to the API if absolutely necessary by creating a custom RBAC role and attaching a service account to that function's deployment.

Pod Security and Function Isolation

OpenFaaS implements several pod-level security best practices out of the box. The OpenFaaS core components (gateway, watchdog, etc.) run as non-root users. Additionally, the official function templates (for Python, Node.js, etc.) use a non-root user to run your code. This means that even if an attacker breaks out of your function code, they have less privilege on the host. You should stick to these templates or ensure your Dockerfiles use non-root users. The Helm chart even has a `setNonRootUser=true` option that forces functions to run as non-root as a safety net.

OpenFaaS does not allow deploying functions with privileged mode by design. Running privileged containers (which have host root access) would be a major security risk. By disallowing it, OpenFaaS ensures no function can unintentionally gain such access.

Support for read-only root filesystems for functions means your function's container filesystem (aside from a `tmp` directory) can be mounted read-only, preventing the function code or libraries from being modified during runtime. A read-only filesystem can be enabled by an annotation or label on the function deployment. This hardening practice is prudent if your function doesn't need to write to disk (or only to

`/tmp`), because it protects against certain tampering and persistence techniques an attacker might use if they manage to exploit a function.
Although not a security feature per se, always set memory and CPU limits for your functions to prevent a malfunctioning or malicious function from consuming excessive resources (which could be a denial-of-service (DoS) vector). OpenFaaS templates often come with some default resource requests and limits in the YAML file. Review and adjust these on the basis of your capacity.

Secure Secrets and Credentials

Functions often need credentials or tokens (for databases, APIs, etc.). OpenFaaS integrates with Kubernetes Secrets to make them secure. Either of the commands

```
faas-cli secret create <name>
kubectl create secret
```

stores secrets in the cluster. For example,

```
echo -n 'my-api-key' | ↩
  faas-cli secret create api-key
```

stores the secret in the *openfaas-fn* namespace by default.

Image Management and Supply Chain

Because OpenFaaS runs Docker images as functions, ensure your images come from a trusted source. If your functions are proprietary, push images to a private registry and use image pull secrets to let OpenFaaS fetch them. Create a Kubernetes Secret of type `docker-registry` and either attach it to the service account or specify it per function. If you use OpenFaaS templates, keep them updated to get security fixes (the templates are essentially base images with the watchdog) and scan `Images` for vulnerabilities as part of your continuous integration and continuous deployment (CI/CD) pipeline.

Conclusion

Deploying OpenFaaS on your machine gives you a flexible, powerful, serverless platform under your control. In this article, I walked you through setting up Kubernetes and OpenFaaS and deploying functions before delving into the critical aspects of security, performance, and scalability.
With these insights, you can confidently deploy OpenFaaS on an Ubuntu machine and in a Kubernetes environment to run serverless functions that meet your organization's needs. As always, continue to refer to the official OpenFaaS documentation [3] and community resources [4] for deeper dives into specific topics or new features. The OpenFaaS project is active and continually improving in areas of developer experience, security, and scalability.

Info

[1] OpenFaaS: [https://www.openfaas.com]
[2] OpenFaaS CE: [https://www.openfaas.com/pricing/]
[3] Docs: [https://docs.openfaas.com]
[4] Community: [https://docs.openfaas.com/community/]

The Author

Marcin Gastol is a Senior DevOps Engineer and a Microsoft MVP and Certified Trainer with extensive experience in Azure technologies. He has taught various IT subjects, hosts a blog that addresses multiple IT areas [https://marcingastol.com/], and speaks at various conferences.

Photo by Toby Elliott on Unsplash

Application-aware batch scheduler

Eruption

Volcano optimizes high-performance workloads on Kubernetes to avoid deadlocks. By Abe Sharp

The basic Kubernetes scheduler – `kube-scheduler` – does a great job of bin-packing pods into nodes, but it can result in a scheduling deadlock for the more complicated multipod jobs that are created by analytics and artificial intelligence and machine learning (AI/ML) frameworks such as Apache Spark and PyTorch. That means expensive cluster resources, such as GPUs, sitting idle and unavailable to any workload.

Volcano scheduler is a Cloud Native Computing Foundation (CNCF) project that introduces the `Queue` and `PodGroup` custom resources to enable gang scheduling (i.e., the simultaneous scheduling of multiple related objects) and facilitate more efficient use of the cluster. Complex jobs run more reliably, and data engineers become more productive.

In this article, I demonstrate default Kubernetes scheduling behavior with the use of short-lived single-pod jobs, show how multipod jobs from Apache Spark and PyTorch can trigger a scheduling lock, and use Volcano to run the same jobs smoothly and predictably. The Git repository [1] gives full details of how to create a test Kubernetes cluster on Digital Ocean and to run all the examples.

Kubernetes for Analytics and ML

Kubernetes, often considered the operating system of the cloud, is often thought of in terms of distributed microservices – in other words, client-server applications with an indefinite lifespan, decomposed into smaller services components (e.g., database, business logic, web front end) for containerized deployment in a way that makes each part redundant, scalable, and easy to upgrade. In that use case, the Kubernetes cluster was most likely designed and scaled with the application's resource requirements in mind, but Kubernetes also lends itself to the "batch" use case – that is, for running resource-intensive jobs of finite duration, particularly in the fields of Big Data analytics, high-performance computing, and training of AI models. In these applications, to make the best use of expensive GPUs, they needs to be shared by multiple users. They do not need to be available to everyone immediately; the users just need to be able to submit their workloads to the cluster and get

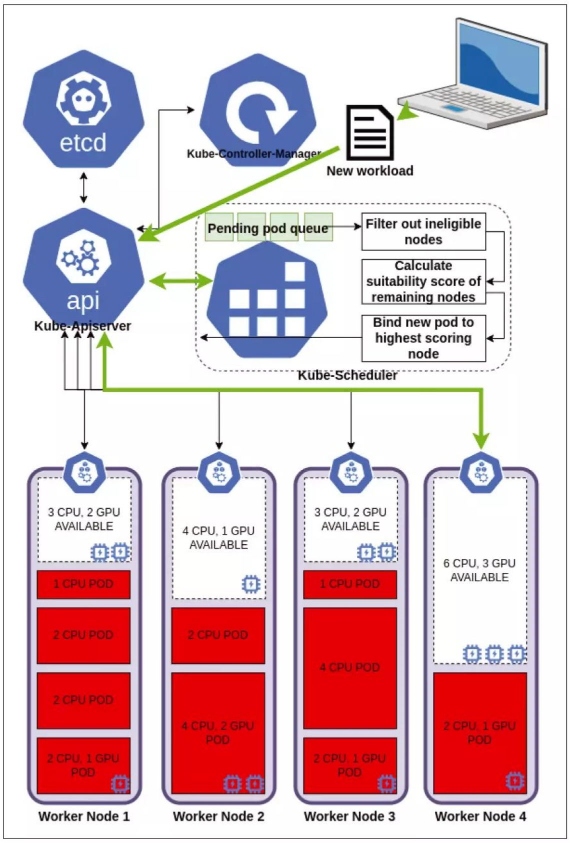

etcd
Kube-Controller-Manager
New workload
api
Kube-Apiserver
Pending pod queue
Filter out ineligible nodes
Calculate suitability score of remaining nodes
Bind new pod to highest scoring node
Kube-Scheduler
3 CPU, 2 GPU AVAILABLE
1 CPU POD
2 CPU POD
2 CPU POD
2 CPU, 1 GPU POD
4 CPU, 1 GPU AVAILABLE
2 CPU POD
4 CPU, 2 GPU POD
3 CPU, 2 GPU AVAILABLE
1 CPU POD
4 CPU POD
2 CPU, 1 GPU POD
6 CPU, 3 GPU AVAILABLE
2 CPU, 1 GPU POD
Worker Node 1
Worker Node 2
Worker Node 3
Worker Node 4

Figure 1: Overview of how `kube-scheduler` filters and scores nodes for pod binding.

the results (e.g., a trained AI model) within a reasonable time frame. Kubernetes' inherent scheduling and bin-packing capabilities – by which it can make efficient use of the compute and memory resources provided by its worker nodes – make it well suited to running such short-lived tasks.

Deploying MinIO

The examples in this article use a large text file (`war-and-peace.txt` from Project Gutenberg [2]) saved in a local Amazon Simple Storage Service (S3)-compatible object storage (Figure 2). This type of object storage, being easy to deploy and access, is a great choice for distributed storage in hybrid environments, where jobs are initiated from a local workstation but executed in a public cloud environment. You can install MinIO with a single `helm` command:

```
helm -n minio install objectstore ↩
  oci://registry-1.docker.io/↩
    bitnamicharts/minio ↩
  --create-namespace
```

However, the Kubernetes implementations of many widely used AI/ML frameworks require the near-simultaneous creation of multiple pods for a job to complete successfully, as the examples in this article will show. If multiple jobs are submitted together, you end up with a situation in which some of the pods from one job are running, some pods from another job are also running, but the cluster has insufficient resources remaining to schedule the remaining pods from either job. As a result, both jobs are indefinitely stuck and the cluster is in a state of "scheduling deadlock."

Scheduling a Kubernetes Pod

A pod is the basic unit of a Kubernetes workload (deriving from a "pod" of containers sharing a common namespace on a node). Its specification includes an array of container images, commands to execute in each, storage and configuration details, and quantities of CPUs, GPUs and memory for each container. The Kubernetes scheduler (`kube-scheduler`), one of the control plane's essential services, decides which node, if any, a pod is scheduled on (or bound to). In Kubernetes terminology, a pod is said to be "scheduled" when it is assigned to a worker node and can be created; it doesn't imply that it can only be run at a particular time, despite what the word implies. (Kubernetes does include a `CronJob` object for that purpose.)

The Kubernetes scheduler operates on a queue of pods that are still in the Pending state (i.e., all the pods that ought to be scheduled, but haven't yet been). That queue is populated by `kube-controller-manager`, another essential control plane service, whose job is to reconcile continually the difference between the declarative state of the cluster's objects and what is actually running on the cluster's nodes in real life.

The scheduler follows a two-step process: (1) It filters the list of nodes to exclude those on which the pod cannot be run. For example, the pod specification requests two CPUs for a container in the pod, so it excludes all nodes that don't have two CPUs worth of free capacity; or, the pod requests a GPU, so the scheduler excludes all nodes that don't have an unused GPU; or, some nodes are

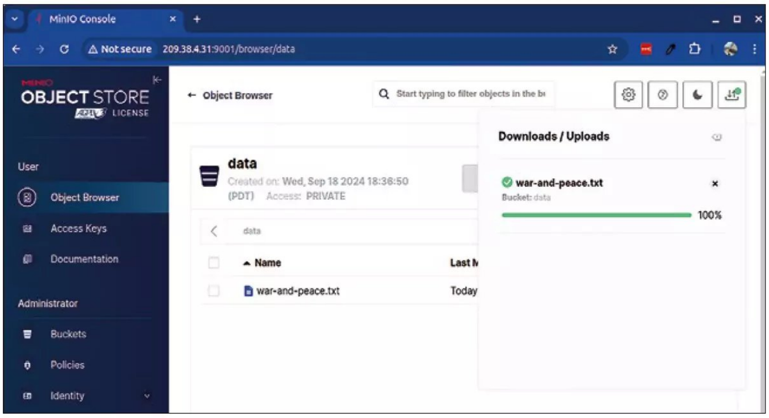

MinIO Console
Not secure 209.38.4.31:9001/browser/data
OBJECT STORE
LICENSE
User
Object Browser
Access Keys
Documentation
Administrator
Buckets
Policies
Identity
Object Browser
Start typing to filter objects in the bu
data
Created on: Wed, Sep 18 2024 18:36:50 (PDT) Access: PRIVATE
data
Name
Last M
war-and-peace.txt
Today
Downloads / Uploads
war-and-peace.txt
Bucket: data
100%

Figure 2: Uploading the text dataset to the MinIO web user interface (UI).

tainted to prevent the scheduling of pods matching a certain specification, so `kube-scheduler` excludes those nodes. (2) It ranks the remaining nodes according to several criteria (e.g., the node that has the most available capacity) and binds the pod to the highest scoring node. Figure 1 illustrates this process, starting with a user submitting a workload to the cluster's API server (`kube-apiserver`).

Limited Resources

In the case of simple batch jobs that each create a single pod, the standard Kubernetes scheduler queues the pending pods and schedules them in turn as resources that are released by the preceding completed pods. I'll demonstrate this by submitting (as simultaneously as possible) 20 two-CPU pods to the test cluster described in the GitHub repo [1]. Each pod runs a Python word count program on the `war-and-peace.txt` file mentioned in the "Deploying MinIO" boxout.
Follow the instructions printed out by the successful Helm deployment to see the MinIO admin password, and then use the

```
kubectl forward
```

command in the same output to access the web UI of your object store. Create a bucket path called `data/events` and upload to it the `war-and-peace.txt` file [2]. Export the MinIO admin password in your local environment as `MINIO_PASSWORD` for later use by the example pods. As a more reliable alternative to `kubectl forward`, change the MinIO service type to make it accessible from outside the cluster via a load balancer and watch the service to discover the public IP address that is assigned to your MinIO service:

```
kubectl ➘
  -n minio patch svc objectstore-minio ➘
  -p '{"spec": {"type": "LoadBalancer"}}'
watch kubectl -n minio get svc -o wide
```

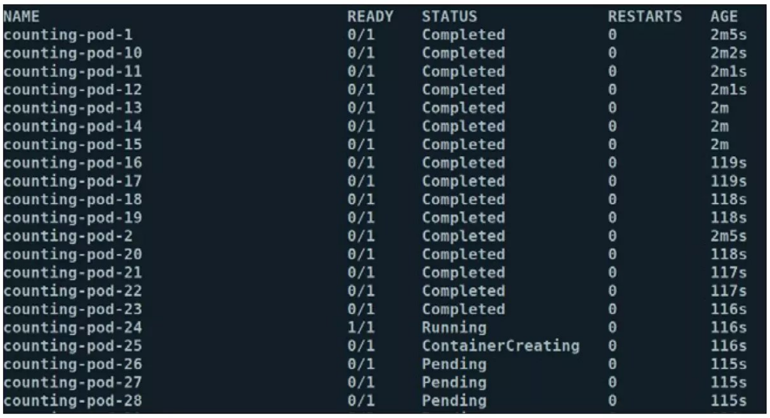
The test cluster has only six CPUs available for workloads, so it obviously won't be able to schedule all the pods at the time they are submitted to the API server. The example directly creates pods for the sake of clarity; however, in a production scenario, you'd probably use a Kubernetes `Job` object, which can be viewed as an overlay to the `Pod` object that allows you to specify the number of attempts that will be made to get the specified pod to run to completion and allow a certain number of historical pod attempts to be retained to allow their inspection of their logs.
The following command downloads the `.yaml` file, submits it to the cluster 20 times (with `envsubst` to create a unique name for each pod and populate the MinIO password into the pod spec), and watches the progress of the resulting pod statuses:

```
curl -LO ➘
  https://raw.githubusercontent.com/➘
  datadoc24/admin-volcano-article/refs/➘
  heads/main/single-pod-wordcount-job/➘
  wordcount-pod.yaml
for n in $(seq 1 20);➘
  do i=$n envsubst < wordcount-pod.yaml ➘
    | kubectl apply -f -;➘
  done
watch kubectl get po
```

In the resulting output shown in Figure 3, you can see each pod transitioning one by one from Pending to Running to Completed. If you want to double-check that your pods really did some useful work, the `kubectl logs` command can be used to show that each pod (even in its Completed state) contains the output of the word count program.

Multipod Analytics and AI

Unlike the simple single-pod example, typical Kubernetes-native AI/ML applications require that several pods be scheduled simultaneously or in a certain order. A good example of this is the Kubernetes implementation of Apache Spark [3], a distributed analytics engine on which users can run SQL-type queries across large datasets. "Distributed" in this case means that a program is controlled by a driver process and split up into a workflow with parallel tasks shared between a number of executor processes. In the Kubernetes context, the driver is one pod, and each executor is another pod, and they all have to run simultaneously for the Spark application to succeed.
Another example is the Kubeflow Training Operator, which is designed to work closely with the PyTorch library to train AI models – neural networks – on large datasets with multiple GPU-enabled pods. Submitting a single PyTorch training job to the Kubernetes cluster requires the cluster to have the requested number of GPUs available for the model training to succeed. I'll give an example of each in turn, to show how creating such jobs can cause a state of scheduling deadlock in the cluster and how the use of the Volcano scheduler can help.



```
NAME                                    READY   STATUS              RESTARTS   AGE
counting-pod-1                          0/1     Completed           0          2m5s
counting-pod-10                         0/1     Completed           0          2m2s
counting-pod-11                         0/1     Completed           0          2m1s
counting-pod-12                         0/1     Completed           0          2m1s
counting-pod-13                         0/1     Completed           0          2m
counting-pod-14                         0/1     Completed           0          2m
counting-pod-15                         0/1     Completed           0          2m
counting-pod-16                         0/1     Completed           0          119s
counting-pod-17                         0/1     Completed           0          119s
counting-pod-18                         0/1     Completed           0          118s
counting-pod-19                         0/1     Completed           0          118s
counting-pod-2                          0/1     Completed           0          2m5s
counting-pod-20                         0/1     Completed           0          118s
counting-pod-21                         0/1     Completed           0          117s
counting-pod-22                         0/1     Completed           0          117s
counting-pod-23                         0/1     Completed           0          116s
counting-pod-24                         1/1     Running             0          116s
counting-pod-25                         0/1     ContainerCreating   0          116s
counting-pod-26                         0/1     Pending             0          115s
counting-pod-27                         0/1     Pending             0          115s
counting-pod-28                         0/1     Pending             0          115s
```

Figure 3: The Kubernetes scheduler schedules single-pod jobs in turn, and no cluster resources are locked.

Scheduling Deadlock

Listing 1: Additions to spark-defaults.confi

```
#VOLCANO
spark.kubernetes.scheduler.name=volcano
spark.kubernetes.scheduler.volcano.podGroupTemplateFile=/home/abe/spark-on-k8s/podgroup-template.yaml
spark.kubernetes.driver.pod.featureSteps=org.apache.spark.deploy.k8s.features.VolcanoFeatureStep
spark.kubernetes.executor.pod.featureSteps=org.apache.spark.deploy.k8s.features.VolcanoFeatureStep
```

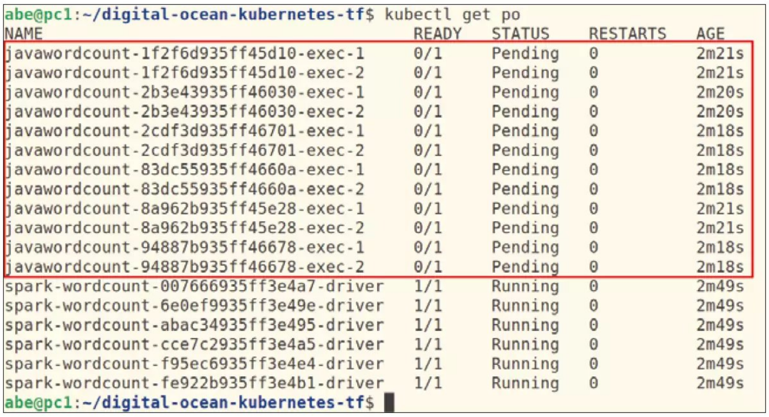
Imagine a team of data scientists running their Apache Spark jobs on a shared Kubernetes (K8s) cluster [3]. On aggregate, the cluster contains ample resources for the whole team, but not enough to run all their jobs simultaneously. Section 3 of the GitHub repository [1] shows how Apache Spark can be downloaded, compiled for use with Kubernetes and Volcano, and containerized. It also shows the JAR files and configuration (`spark-defaults.conf`) needed to enable it for S3 access to work with the MinIO object store used in the single-pod workload example. In this example, I'll run a word count on the same file as before, but as a Spark job distributed between one driver and two executor pods, requesting one CPU each. The first Spark command runs just a single instance of this job:

```
export APISERVER=<K8s API server URL>
spark-submit ➜
  --master k8s://$APISERVER ➜
  --deploy-mode cluster ➜
  --name spark-wordcount ➜
  --class org.apache.spark.examples.➜
    JavaWordCount ➜
    local:///opt/spark/examples/jars/➜
    spark-examples_2.12-3.5.2.jar ➜
    s3a://data/war-and-peace.txt
watch kubectl get po
```

You'll see that the Spark driver pod is created first. After it has been running for a minute or so, it'll create a couple of executor pods (you set the number of executors in `spark-defaults.conf`). The pods will run together for a few seconds more, the executors will terminate, and you'll be left with the driver pod in the Completed state with its logs (again, accessible with `kubectl logs`) showing the results of the word count.

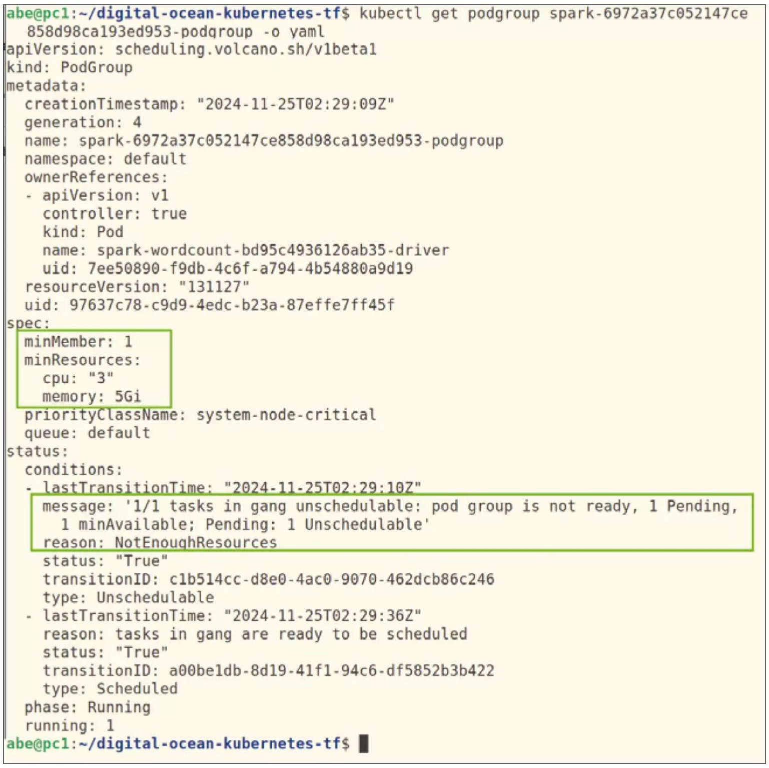
Now, say that six data scientists simultaneously submit Spark jobs, which is simulated with a `for` loop that runs each Spark command as a background process (by suffixing it with the `&` character):

```
for i in $(seq 1 6); do spark-submit ➜
  --master k8s://$APISERVER ➜
  --deploy-mode cluster ➜
  --name spark-wordcount ➜
  --class org.apache.spark.examples.➜
    JavaWordCount ➜
    local:///opt/spark/examples/jars/➜
    spark-examples_2.12-3.5.2.jar ➜
```



```
abe@pc1:~/digital-ocean-kubernetes-tf$ kubectl get po
NAME                                     READY   STATUS    RESTARTS   AGE
javawordcount-1f2f6d935ff45d10-exec-1    0/1     Pending   0          2m21s
javawordcount-1f2f6d935ff45d10-exec-2    0/1     Pending   0          2m21s
javawordcount-2b3e43935ff46030-exec-1    0/1     Pending   0          2m20s
javawordcount-2b3e43935ff46030-exec-2    0/1     Pending   0          2m20s
javawordcount-2cdf3d935ff46701-exec-1    0/1     Pending   0          2m18s
javawordcount-2cdf3d935ff46701-exec-2    0/1     Pending   0          2m18s
javawordcount-83dc55935ff4660a-exec-1    0/1     Pending   0          2m18s
javawordcount-83dc55935ff4660a-exec-2    0/1     Pending   0          2m18s
javawordcount-8a962b935ff45e28-exec-1    0/1     Pending   0          2m21s
javawordcount-8a962b935ff45e28-exec-2    0/1     Pending   0          2m21s
javawordcount-94887b935ff46678-exec-1    0/1     Pending   0          2m18s
javawordcount-94887b935ff46678-exec-2    0/1     Pending   0          2m18s
spark-wordcount-007666935ff3e4a7-driver  1/1     Running   0          2m49s
spark-wordcount-6e0ef9935ff3e49e-driver  1/1     Running   0          2m49s
spark-wordcount-abac34935ff3e495-driver  1/1     Running   0          2m49s
spark-wordcount-cce7c2935ff3e4a5-driver  1/1     Running   0          2m49s
spark-wordcount-f95ec6935ff3e4e4-driver  1/1     Running   0          2m49s
spark-wordcount-fe922b935ff3e4b1-driver  1/1     Running   0          2m49s
abe@pc1:~/digital-ocean-kubernetes-tf$
```

Figure 4: Scheduling deadlock of Apache Spark executor pods.



```
abe@pc1:~/digital-ocean-kubernetes-tf$ kubectl get podgroup spark-6972a37c052147ce
  858d98ca193ed953-podgroup -o yaml
apiVersion: scheduling.volcano.sh/v1beta1
kind: PodGroup
metadata:
  creationTimestamp: "2024-11-25T02:29:09Z"
  generation: 4
  name: spark-6972a37c052147ce858d98ca193ed953-podgroup
  namespace: default
  ownerReferences:
  - apiVersion: v1
    controller: true
    kind: Pod
    name: spark-wordcount-bd95c4936126ab35-driver
    uid: 7ee50890-f9db-4c6f-a794-4b54880a9d19
  resourceVersion: "131127"
  uid: 97637c78-c9d9-4edc-b23a-87effe7ff45f
spec:
  minMember: 1
  minResources:
    cpu: "3"
    memory: 5Gi
  priorityClassName: system-node-critical
  queue: default
status:
  conditions:
  - lastTransitionTime: "2024-11-25T02:29:10Z"
    message: '1/1 tasks in gang unschedulable: pod group is not ready, 1 Pending,
      1 minAvailable; Pending: 1 Unschedulable'
    reason: NotEnoughResources
    status: "True"
    transitionID: c1b514cc-d8e0-4ac0-9070-462dcb86c246
    type: Unschedulable
  - lastTransitionTime: "2024-11-25T02:29:36Z"
    reason: tasks in gang are ready to be scheduled
    status: "True"
    transitionID: a00be1db-8d19-41f1-94c6-df5852b3b422
    type: Scheduled
  phase: Running
  running: 1
abe@pc1:~/digital-ocean-kubernetes-tf$
```

Figure 5: Example of a PodGroup created by Spark, showing that the group requires a total of three CPU resources to run. The group has transitioned from the Unschedulable (Pending) to the Running (Scheduled) state.

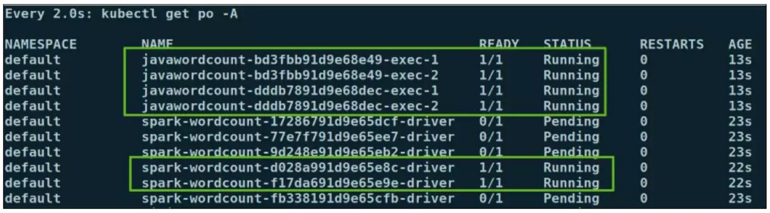
```
    s3a://data/war-and-peace.txt & ↩
  done
watch kubectl get po
```

This example shows that the driver pods of all six jobs start running at about the same time. Each driver is trying to create its two executor pods, but none of them can, because all of the cluster's six available CPUs are consumed by the driver pods themselves (Figure 4).

The drivers keep running for a while before entering the Error state and the Spark jobs just hang until the administrator comes to clean things up. As you can probably tell, I conveniently chose the number of jobs and the pod specifications to trigger this sad state of affairs, but it can easily happen in real life, especially if corporate keeps a close eye on your departmental cloud spend.

Volcano Scheduler

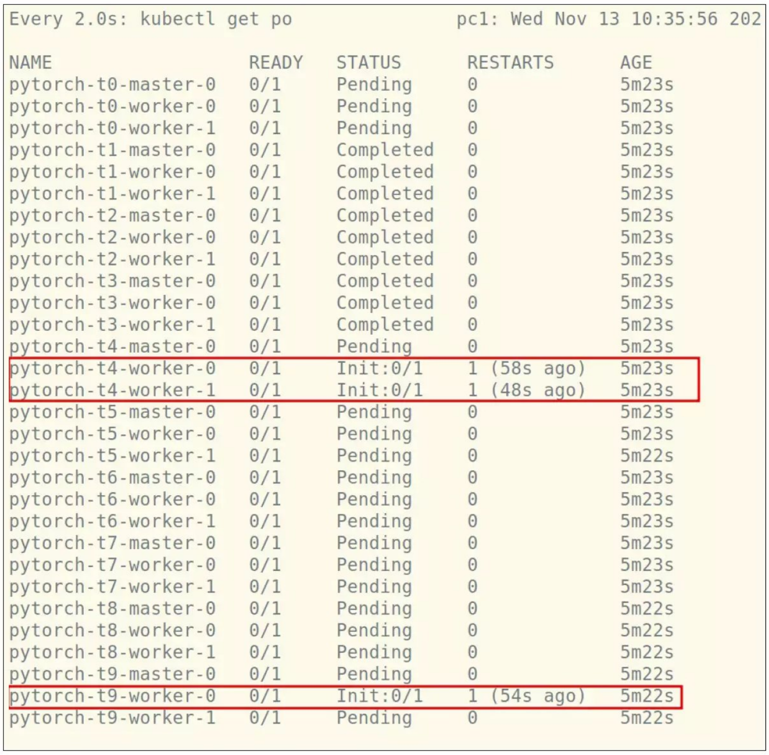
The open source project Volcano [4] was created to solve exactly the problem described here. It is a Kubernetes batch scheduler for workloads that require multiple pods to be scheduled in a timely manner, which it does by creating the custom resources `PodGroup` and `Queue` and implementing them with its own admission controller, ControllerManager, and scheduler that operate in parallel to the standard Kubernetes control plane.



```
Every 2.0s: kubectl get po -A

NAMESPACE           NAME                                        READY   STATUS       RESTARTS   AGE
default             javawordcount-bd3fbb91d9e68e49-exec-1       1/1     Running      0          13s
default             javawordcount-bd3fbb91d9e68e49-exec-2       1/1     Running      0          13s
default             javawordcount-dddb7891d9e68dec-exec-1       1/1     Running      0          13s
default             javawordcount-dddb7891d9e68dec-exec-2       1/1     Running      0          13s
default             spark-wordcount-17286791d9e65dcf-driver     0/1     Pending      0          23s
default             spark-wordcount-77e7f791d9e65ee7-driver     0/1     Pending      0          23s
default             spark-wordcount-9d248e91d9e65eb2-driver     0/1     Pending      0          23s
default             spark-wordcount-d028a991d9e65e8c-driver     1/1     Running      0          22s
default             spark-wordcount-f17da691d9e65e9e-driver     1/1     Running      0          22s
default             spark-wordcount-fb338191d9e65cfb-driver     0/1     Pending      0          23s
```

Figure 6: When configured to use Volcano, Spark will assign all pods for a given Spark job (driver and executors) to the same PodGroup, ensuring they are gang-scheduled. While all cluster resources are occupied with Running pods, all the pods for Pending jobs must wait together.



```
Every 2.0s: kubectl get po                     pc1: Wed Nov 13 10:35:56 202

NAME                  READY   STATUS      RESTARTS       AGE
pytorch-t0-master-0   0/1     Pending     0              5m23s
pytorch-t0-worker-0   0/1     Pending     0              5m23s
pytorch-t0-worker-1   0/1     Pending     0              5m23s
pytorch-t1-master-0   0/1     Completed   0              5m23s
pytorch-t1-worker-0   0/1     Completed   0              5m23s
pytorch-t1-worker-1   0/1     Completed   0              5m23s
pytorch-t2-master-0   0/1     Completed   0              5m23s
pytorch-t2-worker-0   0/1     Completed   0              5m23s
pytorch-t2-worker-1   0/1     Completed   0              5m23s
pytorch-t3-master-0   0/1     Completed   0              5m23s
pytorch-t3-worker-0   0/1     Completed   0              5m23s
pytorch-t3-worker-1   0/1     Completed   0              5m23s
pytorch-t4-master-0   0/1     Pending     0              5m23s
pytorch-t4-worker-0   0/1     Init:0/1    1 (58s ago)    5m23s
pytorch-t4-worker-1   0/1     Init:0/1    1 (48s ago)    5m23s
pytorch-t5-master-0   0/1     Pending     0              5m23s
pytorch-t5-worker-0   0/1     Pending     0              5m23s
pytorch-t5-worker-1   0/1     Pending     0              5m22s
pytorch-t6-master-0   0/1     Pending     0              5m23s
pytorch-t6-worker-0   0/1     Pending     0              5m23s
pytorch-t6-worker-1   0/1     Pending     0              5m23s
pytorch-t7-master-0   0/1     Pending     0              5m23s
pytorch-t7-worker-0   0/1     Pending     0              5m23s
pytorch-t7-worker-1   0/1     Pending     0              5m23s
pytorch-t8-master-0   0/1     Pending     0              5m22s
pytorch-t8-worker-0   0/1     Pending     0              5m22s
pytorch-t8-worker-1   0/1     Pending     0              5m22s
pytorch-t9-master-0   0/1     Pending     0              5m22s
pytorch-t9-worker-0   0/1     Init:0/1    1 (54s ago)    5m22s
pytorch-t9-worker-1   0/1     Pending     0              5m22s
```

Figure 7: The first few jobs completed successfully, but because of no coordination at the job level, pods with GPU resources soon become split between partially deployed jobs and hang indefinitely.

When you create a PodGroup, you specify the minimum number of member pods and the minimum amount of resources that the cluster must be able to provide for any pod labeled as being a member of the group to be scheduled. None of the pods in the group will be scheduled until the cluster can provide the resources specified in the PodGroup (in terms of compute resources and pod numbers), and PodGroups can be added to any chosen Queue object; Queues can be assigned different priorities to give further control over how resources are allocated. The following commands install Volcano with Helm and show the new custom resources created:

```
helm repo add volcano-sh ↩
  https://volcano-sh.github.io/helm-charts
helm repo update
helm install volcano volcano-sh/volcano ↩
  -n volcano-system --create-namespace
kubectl api-resources | grep volcano
```

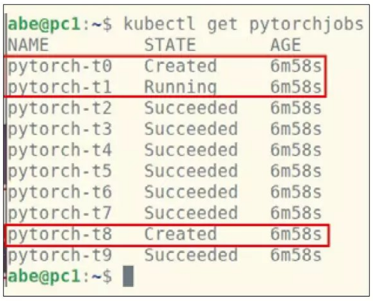
Listing 1 shows the relevant additions made to the `spark-defaults.conf` file to configure Spark to use Volcano as its scheduler. The complete file is available in the project Git repository [1]. With this configuration, Spark's Kubernetes integration will create all pods for a certain job as members of the same PodGroup, allowing Volcano to gang-schedule them together and avoid the deadlock situation created by partially created jobs.

When you rerun the `spark-submit` command, it's exactly the same as before, only now you are leveraging Volcano, and all the jobs are executed quickly. Figure 5 shows the details of one of the PodGroups, and Figure 6 shows all of the pods that were created. It's not clear from the pod names, but each of the two running Spark driver pods has two running executor pods. The other driver pods are pending, and the remaining executor pods have not been created at all (because they are created by their associated driver pod).

Training a PyTorch Model

The second example of a resource-intensive job comes from PyTorch, which is a Python machine learning library that (among many other capabilities) simplifies the training of ML models on CPU or GPU hosts. The Kubeflow Kubernetes machine learning platform offers a particular feature called the Kubeflow Training Operator [5] with an accompanying PyTorchJob custom resource that is designed to train models on behalf of PyTorch programs, with the use of the resources of a Kubernetes cluster. This use case is well suited to a team of data scientists who want to run Python code on their local machines but use a shared group of GPU resources to train their models as fast as possible.

PyTorch spreads training across multiple hosts (and therefore GPUs) with its `DistributedDataParallel` (DDP) module. When run with the Kubeflow Training Operator, it necessarily translates to multiple pods running simultaneously. The Kubeflow Training Operator is designed to work with Volcano to provide gang-scheduling so that if the cluster doesn't have sufficient resources to set up all of the training job's resources, then none will be scheduled at all, and the PodGroup will remain in the Queue until such a time as the resources do become available.



```
abe@pc1:~$ kubectl get pytorchjobs
NAME          STATE       AGE
pytorch-t0    Created     6m58s
pytorch-t1    Running     6m58s
pytorch-t2    Succeeded   6m58s
pytorch-t3    Succeeded   6m58s
pytorch-t4    Succeeded   6m58s
pytorch-t5    Succeeded   6m58s
pytorch-t6    Succeeded   6m58s
pytorch-t7    Succeeded   6m58s
pytorch-t8    Created     6m58s
pytorch-t9    Succeeded   6m58s
abe@pc1:~$
```

Figure 8: The PyTorchJobs corresponding to the pods in Figure 7 are stuck in this state.

Running in its default state, without Volcano, multiple PyTorchJobs can each take up all available cluster GPUs without any one of them having enough to train their model properly, leading to another deadlock situation.

In this example, the test cluster is recreated with the `gpupool` section of the `doc8s.tf` OpenTofu module ([1], section 1) uncommented.

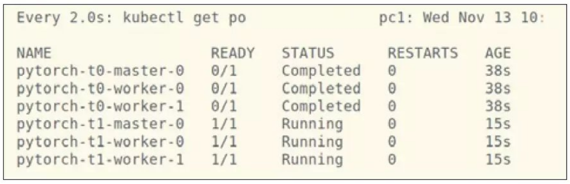
To create six PyTorchJobs simultaneously with Python threads, install the Kubeflow Training Operator [5] and run the `mnist-gpu.py` script from section 4 of the GitHub repo [1], which is a slightly modified version of the example given in the Kubeflow Training Operator documentation:

```
kubectl apply ➘
  -k "github.com/kubeflow/➘
    training-operator.git/manifests/➘
    overlays/standalone?ref=master"
python mnist-gpu.py
```

As Figure 7 and Figure 8 show, you hit a deadlock situation with GPU hosts tied up by different partially created jobs.

To leverage Volcano in the Kubeflow Training Operator [6], edit the `training-operator` deployment to set the gang-scheduler command-line option (Listing 2).

After retrying the training example, you can see that the PyTorchJobs are created one after the other so that each one can consume the GPU resources it needs (Figure 9).



```
Every 2.0s: kubectl get po                    pc1: Wed Nov 13 10:

NAME                    READY   STATUS      RESTARTS   AGE
pytorch-t0-master-0     0/1     Completed   0          38s
pytorch-t0-worker-0     0/1     Completed   0          38s
pytorch-t0-worker-1     0/1     Completed   0          38s
pytorch-t1-master-0     1/1     Running     0          15s
pytorch-t1-worker-0     1/1     Running     0          15s
pytorch-t1-worker-1     1/1     Running     0          15s
```

Figure 9: Even though the jobs *pytorch-t0* through *pytorch-t5* have all been submitted to the cluster, Volcano doesn't even begin to create the pods for a particular job until it can be sure sufficient GPU hosts are available.

Listing 2: Edited training-operator

```
kubectl -n kubeflow edit deploy training-operator
spec:
  containers:
    -- command:
      -- /manager
+     -- --gang-scheduler-name=volcano
     image: kubeflow/training-operator
     name:  training-operator
```

Conclusion

Shared compute resources are expensive, whether they are private or in the public cloud, and nobody wants their jobs to be delayed indefinitely because of deadlock caused by scheduling limitations. In this article, I described how Volcano schedules PodGroups and demonstrated how easily it can be used by common open source AI/ML frameworks to get the most out of Kubernetes worker hosts. ■

Info

[1] Examples in this article: [https://github.com/datadoc24/admin-volcano-article/tree/main]

[2] *War and Peace* from Project Gutenberg: [https://www.gutenberg.org/ebooks/2600]

[3] Apache Spark on Kubernetes: [https://spark.apache.org/docs/latest/running-on-kubernetes.html#using-volcano-as-customized-scheduler-for-spark-on-kubernetes]

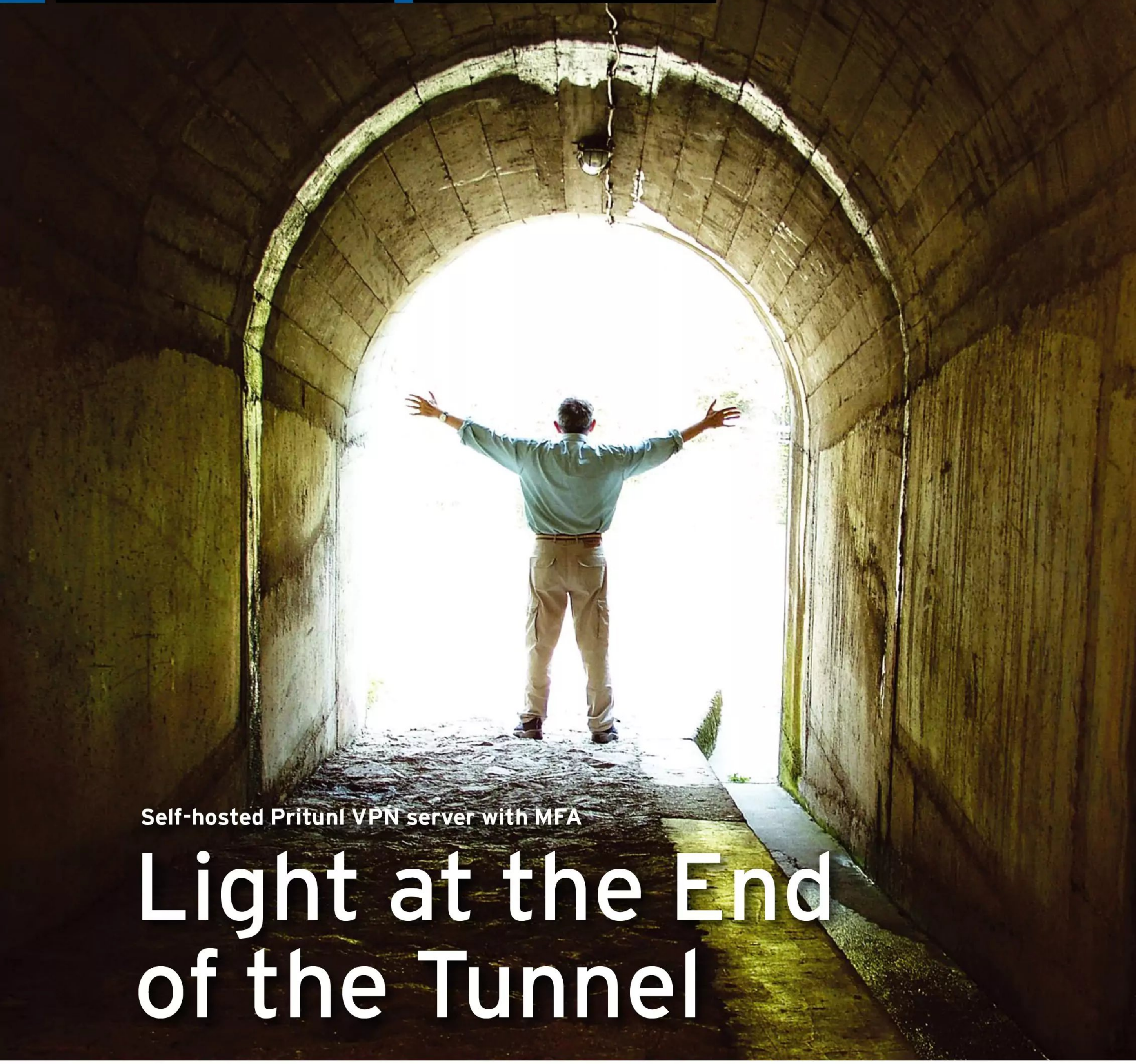
[4] Volcano scheduler: [https://volcano.sh/en/docs/]

[5] Kubeflow Training Operator: [https://www.kubeflow.org/docs/components/training/getting-started/]

[6] Enabling Volcano gang scheduling for Kubeflow Training Operator: [https://www.kubeflow.org/docs/components/training/user-guides/job-scheduling/#running-jobs-with-gang-scheduling]

Author

Abe Sharp is a Distinguished Technologist at Hewlett Packard Enterprise, where he works on containerized AI and MLOps platforms in the Ezmeral business unit.



Lead Image © Darko Novakovic, fotolia.com

Self-hosted Pritunl VPN server with MFA

Light at the End of the Tunnel

We give you the ultimate guide to deploying a self-hosted Pritunl VPN server with multifactor authentication.

By Conor Fitzgerald

A virtual private network (VPN) can typically be described as a solution that allows an encrypted secure connection, often referred to as a tunnel, to be established to a private network. Many people associate a VPN with remote working, whereby an employee would establish a VPN connection to their corporate network (e.g., from their home office), allowing them to access resources such as file shares, intranet sites, and internal applications securely.

Although the prevalence of cloud solutions have certainly reduced the need for VPNs, deploying a VPN solution can be considered for many other reasons. In addition to secure remote access, VPN solutions offer enhanced security on free public WiFi, and they can also circumvent geo-restricted content by routing all Internet traffic over the VPN tunnel. Furthermore, a VPN can be used by an organization to provide site-to-site connectivity between remote offices, thus allowing that organization to share resources while giving the end users the convenience of being on the same logical network. Although many commercial VPN solutions are on the market, several open source self-hosted alternatives are available, and one of the best open source VPN solutions to deploy across your corporate or home network is Pritunl. The benefits of a Pritunl VPN include:

- Open Source – Pritunl is free to use with no set limit on the number of users.

- Secure – Pritunl uses strong encryption protocols and supports multifactor authentication (MFA).
- Scalable – Pritunl can be deployed to handle a large number of VPN users at any one time.
- Ease of Use – Pritunl is a user-friendly VPN solution.
- Multiple Use Cases – Pritunl provides secure remote access along with site-to-site connectivity to establish VPN connectivity between multiple office locations.

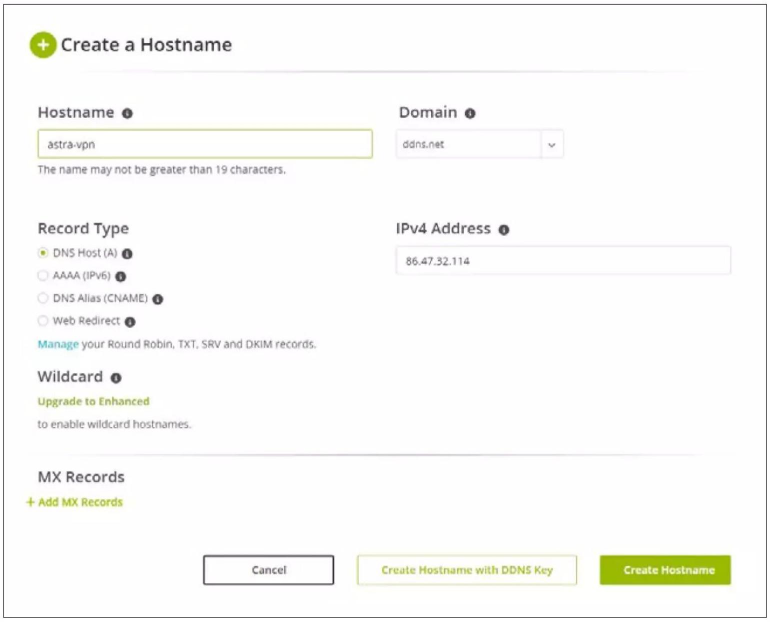
In this article, I show you how to deploy a single-instance Pritunl VPN server running Ubuntu 24.04.01 LTS, configure a No-IP dynamic DNS hostname to use with your VPN, configure port forwarding on your router to allow remote access client connectivity to the Pritunl VPN server, and configure a VPN user account that uses Google Authenticator as part of the secure MFA process. It is important to note that all components of this solution are open source, so it really does offer an enterprise-class VPN solution for organizations with very limited IT budgets.

Set Up Dynamic DNS

One of the main requirements when setting up a self-hosted VPN service is to know the public IP address of the network where the VPN server is to be hosted. The VPN clients will use that public IP address to connect to the VPN. Unfortunately, the majority of home or small business Internet connections will get a dynamic public IP address from their ISP, meaning that the public address of your Internet connection will change on a frequent basis. The good news is that a dynamic DNS (DDNS) service allows you to create a user-friendly DNS hostname against which you map your dynamic public IP address. A small updater agent or client runs inside your network to update the dynamic DNS hostname each time your public IP address changes. This client can be installed on Windows, Mac, and Linux and can often be found as an inbuilt client on your home router, typically in an advanced configuration area under *Dynamic DNS* or *DDNS*.

No-IP is one company that provides DNS and domain services. No-IP offers a free dynamic DNS service that is limited to updating one hostname. The first thing you need to do is create a free account by visiting the No-IP website [1] and signing up. Once your account is created, you need to create a hostname; in this example, I use *astra-vpn.ddns.net* (Figure 1).

At this stage you should configure the dynamic DNS update client on



Create a Hostname

Hostname: astra-vpn

The name may not be greater than 19 characters.

Domain: ddns.net

Record Type

● DNS Host (A)

○ AAAA (IPv6)

○ DNS Alias (CNAME)

○ Web Redirect

Manage your Round Robin, TXT, SRV and DKIM records.

IPv4 Address: 86.47.32.114

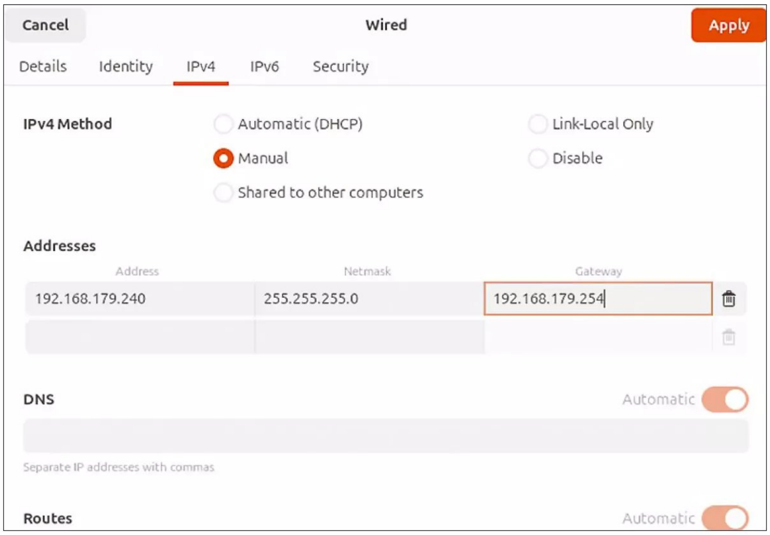
Wildcard

Upgrade to Enhanced

to enable wildcard hostnames.

MX Records

Figure 1: Creating a public hostname to use with your VPN.



IPv4 Method

○ Automatic (DHCP)

● Manual

○ Shared to other computers

○ Link-Local Only

○ Disable

Addresses

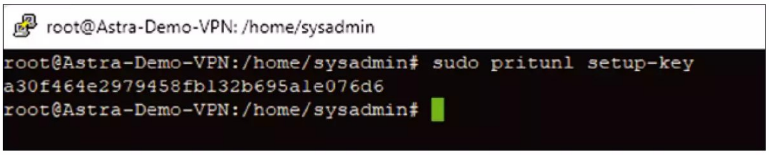
Address	Netmask	Gateway
192.168.179.240	255.255.255.0	192.168.179.254

DNS: Automatic

Separate IP addresses with commas

Routes: Automatic

Figure 2: Configuring a static IP address on a VPN server.



root@Astra-Demo-VPN: /home/sysadmin

```
root@Astra-Demo-VPN:/home/sysadmin# sudo pritunl setup-key
a30f464e2979458fb132b695a1e076d6
root@Astra-Demo-VPN:/home/sysadmin# 
```

Figure 3: Retrieving the database setup key required for the initial configuration.

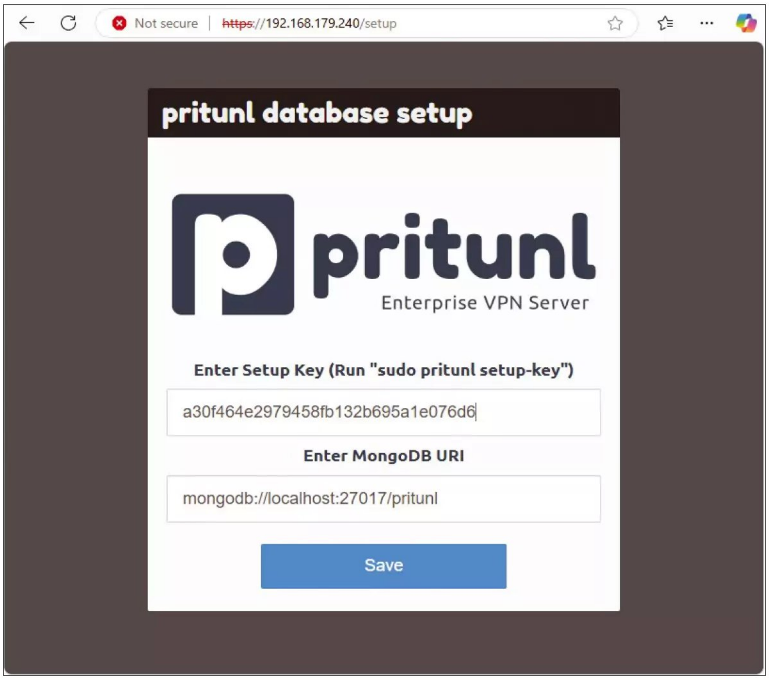
the device of your choice within your network. You can also choose to run the dynamic DNS update client for Linux on the VPN server (see the “Install the No-IP Linux DUC on Ubuntu” box).

Install the Pritunl VPN Self-Hosted Server

With your dynamic DNS hostname registered, you can now move to the next step, which is to install the Pritunl VPN server. For the purposes here, assume that a new virtual machine has been set up on the hypervisor of your choice running a fresh installation of Ubuntu Desktop 24.04.01 LTS [2]. The Ubuntu virtual machine needs to be configured to use a static IP address. In this example, the network is 192.168.179.0/24, and I have configured the Ubuntu virtual machine to use a static IP address of 192.168.179.240 (Figure 2).
Other prerequisites include setting a root password, running updates on the Ubuntu virtual machine, installing SSH, temporarily disabling the firewall, and installing utilities (e.g., the `curl` utility and the *net-tools* package) from the terminal in the Ubuntu virtual machine:

```
sudo passwd root
sudo apt-get update && apt-get upgrade
sudo ufw disable
sudo apt-get install openssh-server
sudo apt-get install net-tools
sudo apt-get install curl
```

At this point, you are ready to install the Pritunl VPN software, so visit Pritunl [3] and click the *Get Started* box toward the bottom of the page. From there, you are going to run a series of commands taken directly from the official Pritunl installation guide for Ubuntu 24.04.01 LTS (click the *Other Providers Ubuntu 24.04* link in the right margin). Connect to your server over SSH with Putty and,


Not secure | ~~https~~://192.168.179.240/setup

pritunl database setup

Enter Setup Key (Run "sudo pritunl setup-key")
a30f464e2979458fb132b695a1e076d6

Enter MongoDB URI
mongodb://localhost:27017/pritunl

Figure 4: Inputting the database setup key as part of the initial configuration.

Listing 1: Install Printunl VPN Software

```
# Add the repository for MongeDB
sudo tee /etc/apt/sources.list.d/mongodb-org.list << EOF
deb [ signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg ]
    https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse
EOF

# Add the repository for OpenVPN
sudo tee /etc/apt/sources.list.d/openvpn.list << EOF
deb [ signed-by=/usr/share/keyrings/openvpn-repo.gpg ]
    https://build.openvpn.net/debian/openvpn/stable noble main
EOF

# Add the repository for Pritunl
sudo tee /etc/apt/sources.list.d/pritunl.list << EOF
deb [ signed-by=/usr/share/keyrings/pritunl.gpg ]
    https://repo.pritunl.com/unstable/apt noble main
EOF

# Add gnupg
sudo apt --assume-yes install gnupg

# Download the GPG keys for MongoDB, OpenVPN, and Pritunl
curl -fsSL https://www.mongodb.org/static/pgp/server-7.0.asc | sudo gpg -o /
    usr/share/keyrings/mongodb-server-7.0.gpg --dearmor --yes
curl -fsSL https://swupdate.openvpn.net/repos/repo-public.gpg | sudo gpg -o /
    usr/share/keyrings/openvpn-repo.gpg --dearmor --yes
curl -fsSL https://raw.githubusercontent.com/pritunl/pgp/master/pritunl_repo_
    pub.asc | sudo gpg -o /usr/share/keyrings/pritunl.gpg --dearmor --yes

# Update the APT package list
sudo apt update

# Install the packages for Pritunl, MongoDB, and Wireguard
sudo apt --assume-yes install pritunl mongodb-org wireguard-tools

# Start the newly installed MongoDB and Pritunl services
sudo systemctl start mongod pritunl

# Enable the required services to start each time the VPN server reboots
sudo systemctl enable mongod pritunl
```

with root priviledges, run the commands shown in Listing 1.

The gnupg tool verifies the authenticity of downloaded packages and is necessary to handle GNU Privacy Guard (GPG) keys used as part of the Pritunl installation. Updating the APT package list ensures the Ubuntu operating system is aware of the newly added repositories.

At this stage the self-hosted Pritunl VPN server is installed, and you can move to the configuration stage.

Configure the Pritunl VPN Server

The first configuration task when completing the initial setup of the Pritunl VPN server is to complete the Pritunl database setup. To do so, connect to the server over SSH and run

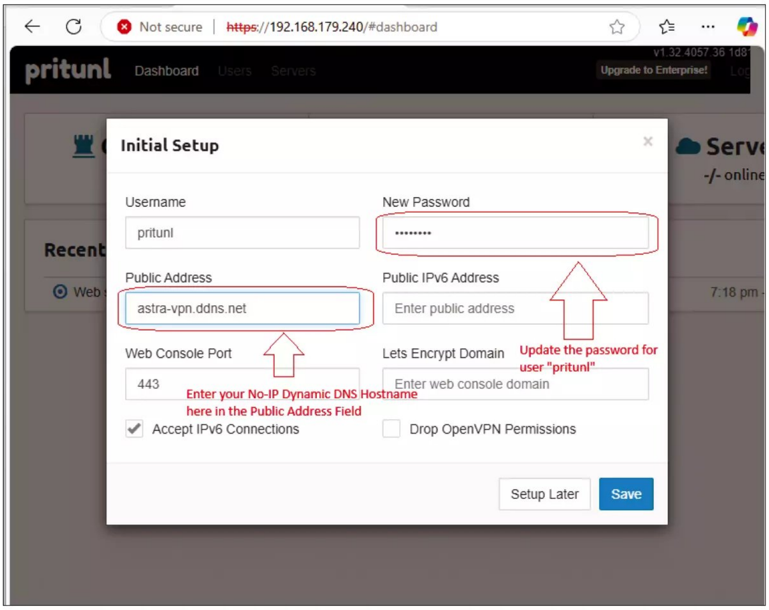
```
sudo pritunl setup-key
```

taking note of the generated key (Figure 3). At this stage, the web-based management console for the VPN server will be available in your browser at *https://your-servers-ip-address*. For this example, the web interface is *https://192.168.179.240*; at this point, you can input the database setup key and click *Save* (Figure 4); then, follow the instructions to retrieve the initial logon information.

As part of the initial setup you need to update the public address and the default password for user *pritunl*, which is the default administration account for the Pritunl VPN server. Update as per the example in Figure 5, remembering to use the No-IP dynamic DNS hostname you configured for use with your VPN. Once done, click *Save*.

With the public address configured and the default password changed, the next step is to configure an organization by clicking *Users* | *Add Organization* (Figure 6). Choose a relevant organization name, enter the details, and click *Add*.

Next, configure the server and attach it to the organization (Figure 7). As per the example here, you need to configure a name, and at this stage, you can


Not secure
pritunl
Dashboard
Users
Servers
v1.32.4057.36 1d8
Upgrade to Enterprise!
Serve
-/- online
Recent
Web
7:18 pm
Initial Setup
Setup Later
Save

https://192.168.179.240/#dashboard

Username: pritunl

New Password: ••••••••

Public Address: astra-vpn.ddns.net

Public IPv6 Address: Enter public address

Web Console Port: 443

Lets Encrypt Domain: Enter web console domain

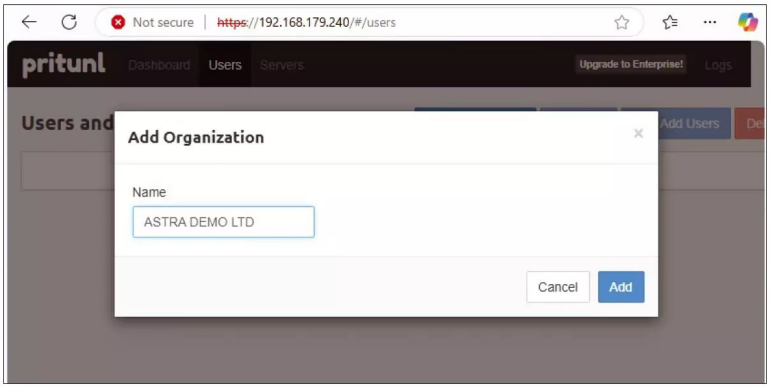
☑ Accept IPv6 Connections

☐ Drop OpenVPN Permissions

Update the password for user "pritunl"

Enter your No-IP Dynamic DNS Hostname here in the Public Address Field

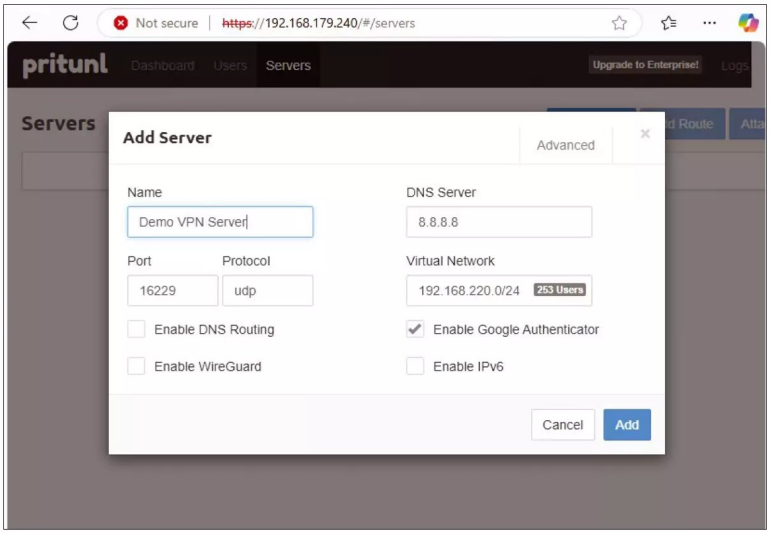
Figure 5: Changing the default administration password and setting the public address.


Not secure
pritunl
Dashboard
Users
Servers
Upgrade to Enterprise!
Logs
Users and
Add Users
Add Organization
Cancel
Add

https://192.168.179.240/#/users

Name: ASTRA DEMO LTD

Figure 6: Adding an organization name.


Not secure
pritunl
Dashboard
Users
Servers
Upgrade to Enterprise!
Logs
Servers
d Route
Atta
Add Server
Advanced
Cancel
Add

https://192.168.179.240/#/servers

Name: Demo VPN Server

DNS Server: 8.8.8.8

Port: 16229

Protocol: udp

Virtual Network: 192.168.220.0/24 253 Users

☐ Enable DNS Routing

☑ Enable Google Authenticator

☐ Enable WireGuard

☐ Enable IPv6

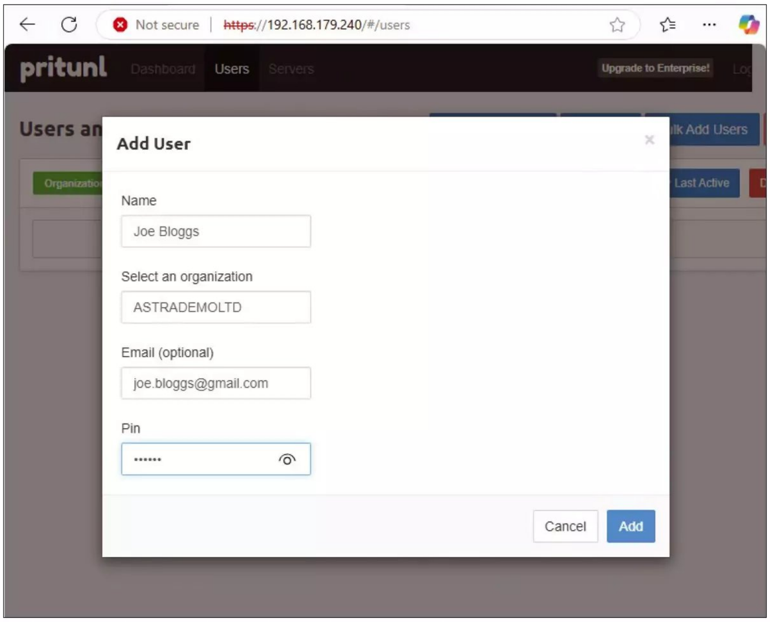
Figure 7: Adding a server configuration as part of the initial VPN installation.

decide whether you want to switch on multifactor authentication with Google Authenticator by ticking the checkbox to *Enable Google Authenticator* before clicking *Add*. Take note of the Port and Protocol fields in this dialog. On your Internet router, you need to create a port forward to send all VPN clients to the internal IP address of your Pritunl VPN server. Please be aware that this port might be different on your self-hosted Pritunl VPN server setup. In this example, the port is *16229* and uses the *udp* protocol.

You will be prompted to attach the organization to the server. At this stage, the initial configuration of your Pritunl VPN server is complete, and you can now start to create VPN user accounts and issue VPN profiles and soft tokens to your end users.

VPN User Accounts and Soft Tokens

Creating a new VPN user account and soft token is really easy. Under *Users* | *Add User*, complete the basic form and click *Add* (Figure 8). Once the user account is created, within the *Settings* tab, you will be able to


Not secure
https://192.168.179.240/#/users
pritunl
Dashboard
Users
Servers
Upgrade to Enterprise!
Bulk Add Users
Last Active
Organization

Add User

Name
Joe Bloggs

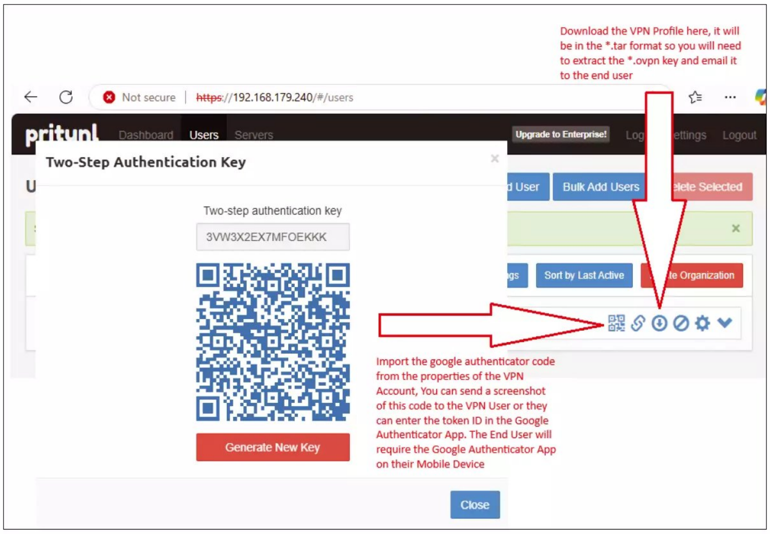
Select an organization
ASTRADEMOLTD

Email (optional)
joe.bloggs@gmail.com

Pin
••••••

Cancel
Add

Figure 8: Adding your first Pritunl VPN user account.


Download the VPN Profile here, it will be in the *.tar format so you will need to extract the *.ovpn key and email it to the end user

Not secure
https://192.168.179.240/#/users
pritunl
Dashboard
Users
Servers
Upgrade to Enterprise!
Logout
Bulk Add Users
Sort by Last Active

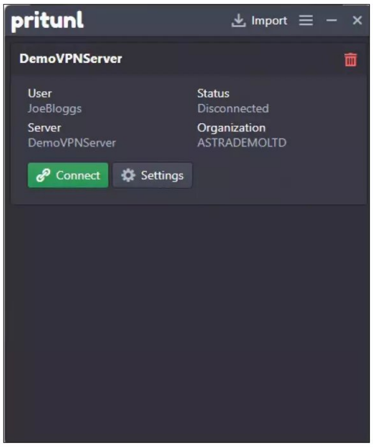
Two-Step Authentication Key

Two-step authentication key
3VW3X2EX7MFOEKKK

Generate New Key
Close

Import the google authenticator code from the properties of the VPN Account, You can send a screenshot of this code to the VPN User or they can enter the token ID in the Google Authenticator App. The End User will require the Google Authenticator App on their Mobile Device

Figure 9: Downloading the VPN token and the Google Authenticator key.

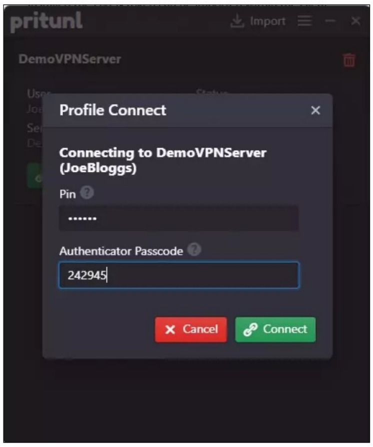

pritunl
Import

DemoVPNServer

User
JoeBloggs
Status
Disconnected
Server
DemoVPNServer
Organization
ASTRADEMOLTD

Connect
Settings

Figure 10: How to connect to the Pritunl VPN server with the Pritunl VPN client.


Profile Connect

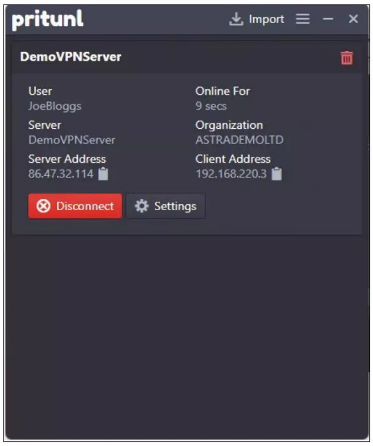
Connecting to DemoVPNServer (JoeBloggs)

Pin
••••••

Authenticator Passcode
242945

Cancel
Connect

Figure 11: Prompting for multifactor authentication on connecting to the VPN.


pritunl
Import

DemoVPNServer

User
JoeBloggs
Online For
9 secs
Server
DemoVPNServer
Organization
ASTRADEMOLTD
Server Address
86.47.32.114
Client Address
192.168.220.3

Disconnect
Settings

Figure 12: A successful connection to the VPN server.

download the VPN token, referred to as the *Profile*, and obtain the Google Authenticator MFA token associated with the account.

Connect to the VPN

To connect to the Pritunl VPN, the VPN profile token needs to be imported into the VPN client, and the Google Authenticator code needs to be imported into the Google Authenticator app. The VPN administrator can download both tokens from the admin console on the VPN server (Figure 9). The VPN profile token will download in the `*.tar` format, and the `*.ovpn` token will need to be extracted from the TAR file and sent to the email address of the end user. A screenshot of the Google Authenticator authentication key can also be emailed to the end user.

Install the No-IP Linux DUC on Ubuntu

The No-IP dynamic DNS update client (DUC) can be run on the same Ubuntu server as the Pritunl VPN. To do so, go to the No-IP sign-in page [5], log in to your No-IP account, and under *Dynamic Update Client* select the *Linux* configuration information (Figure 13).

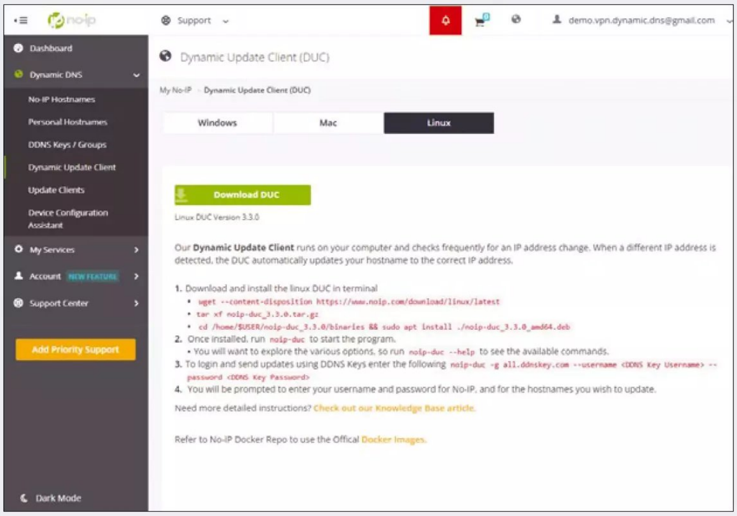
From an SSH client such as Putty, connect to your Ubuntu VPN server, and as root, run

```
wget --content-disposition https://www.noip.com/download/linux/latest
tar xf noip-duc_3.3.0.tar.gz
cd /home/$USER/noip-duc_3.3.0/binaries && sudo apt install ./noip-duc_3.3.0_amd64.deb
```

to install the No-IP Linux dynamic DNS updater client. Finally, run

```
noip-duc -g all.ddnskey.com --username <DDNS key username> --password <DDNS key password>
```

with your specific username and password for your No-IP account.



Linux DUC Version 3.3.0

Our **Dynamic Update Client** runs on your computer and checks frequently for an IP address change. When a different IP address is detected, the DUC automatically updates your hostname to the correct IP address.

1. Download and install the linux DUC in terminal
   - `wget --content-disposition https://www.noip.com/download/linux/latest`
   - `tar xf noip-duc_3.3.0.tar.gz`
   - `cd /home/$USER/noip-duc_3.3.0/binaries && sudo apt install ./noip-duc_3.3.0_amd64.deb`
2. Once installed, run `noip-duc` to start the program.
   - You will want to explore the various options, so run `noip-duc --help` to see the available commands.
3. To login and send updates using DDNS Keys enter the following `noip-duc -g all.ddnskey.com --username <DDNS Key Username> --password <DDNS Key Password>`
4. You will be prompted to enter your username and password for No-IP, and for the hostnames you wish to update.

Need more detailed instructions? Check out our Knowledge Base article.

Refer to No-IP Docker Repo to use the Offical Docker Images.

Figure 13: Installing the No-IP update client on your VPN server.

After downloading and installing the Windows client for the Pritunl VPN [4], the OVPN token can be imported within the client (Figure 10). On initiating the connection to the Pritunl VPN server, the user will be prompted for their pin and the Google Authenticator code (Figure 11). Clicking *Connect* then connects to the Pritunl VPN.

Once connected to the VPN, the end user will be able to see the length of time they have been connected, their VPN client address, and other information (Figure 12). At this stage, the VPN connection is established, and the end user can access any resources (e.g., file shares, printers, internal web applications, etc.).

The VPN administrator can also monitor connections to the VPN server from the admin console, where you can see which VPN users are connected to the VPN in real time. ■

Info

[1] No-IP dynamic DNS account: [https://www.noip.com/]

[2] Ubuntu Desktop: [https://ubuntu.com/desktop]

[3] Pritunl: [https://pritunl.com]

[4] Pritunl client: [https://client.pritunl.com]

[5] No-IP dynamic DNS update client: [https://my.noip.com/dynamic-dns/duc]

Author

Conor Fitzgerald is an IT professional with over 20 years of knowledge and expertise. He is keenly interested in all aspects of IT and is a huge advocate of finding real-world solutions to modern day technological challenges.

■

Photo by Oli Zubenko on Unsplash

Simplify your migration from Docker to Podman

Contained

Podman has become an attractive alternative to Docker, not least because it does not require a central daemon. In this article, we look at how to set about changing the guard and how you can use Podman Desktop to do so.

By Holger Reibold

Commercialization of open source projects and haggling over licensing issues are gnawing away at the acceptance of free software. This not uncommon switch to a commercial future often sounds the death knell for the community's willingness to participate in ongoing development. An unclear licensing policy on the part of Docker developers has led to many controversies in the past, and an audit clause now imposes an additional burden on IT departments. Little wonder then that Red Hat, SUSE, and other Linux distributions replaced Docker with Podman years ago.

To future-proof your own IT infrastructure, you need to be able to plan for the long term. And long discussions about licenses are likely to be counterproductive. Enter Podman [1], the open source project that has long been touted in the community as a replacement for Docker. A number of aspects – both strategic and technological – speak in favor of switching to Podman, including Podman Desktop, which simplifies container management.

Licensing Hassle at Docker

What kicked off the far-reaching changes to the licensing policy was the acquisition of Docker Inc. by Mirantis in November 2019. The provider now relies on a subscription model: For companies with more than 250 employees or an annual turnover of more than $10 million, depending on the subscription option chosen, the fee is $5, $9, or $24 per user per month – all with annual payment. Depending on how you use Docker in your organization, this can mean substantial additional IT costs.

The audit clause is another pain. According to the license agreements, users grant Docker the right to store data relevant to licensing for at least three years, which means that organizations need to take care to archive the data in a reliable way. The use of a special software asset management environment is required to provide protection against potential contract penalties. The personal version is still available for free use, however.

Understandably, the new owners of Docker want and need to earn money with their investment to drive future development, which in itself reveals a significant strategic difference.

Although Podman is backed by a well-known industry player (Red Hat), their business model focuses on generating revenue with licenses and support from its Red Hat Enterprise Linux (RHEL) and OpenShift businesses.

Podman vs. Docker

Fundamentally, Podman and Docker offer the same kind of scalable execution, management, and provisioning of containers. At the same time, the environments have some fundamental differences. What is perhaps the most significant difference relates to the architecture. Whereas Docker is based on a client-server model, Podman has an architecture that does not rely on a daemon, which removes the need to run an uninterrupted daemon process in the background.

The benefits of direct management are a far faster launch of Podman containers. Some sources report performance gains of up to 50 percent. Podman enhances security in the environment, as well.

More Secure Out of the Box

When Docker launches a container, the Docker client sends a request to the Docker daemon, which in turn is responsible for starting the container. Therefore, containers are child processes of the Docker daemon. If a security-related incident originating from a container process occurs and is registered by the Linux audit system (`auditd`), the audit user ID comes up as invalid, instead of the user ID that started the misbehaving container that is evaluated as critical.

As a result, it is almost impossible to assign critical activities to a specific user, which has a significant effect on system security. The situation with Podman is different. Every container is instantiated directly by a user login session. The source information is kept in the container process data and `auditd` can map the user ID to specific container processes, preserving the audit trail.

The idea behind containerized environments is to isolate their execution from the host system as far as possible. The greatest risk is posed by attackers breaking out of a container and compromising the host system.

To mitigate this kind of risk, the containers are executed with minimal permissions. Many administrators and developers appreciate that the container environment strictly counteracts these risks. In principle, Docker can also handle containers without root privileges.

By default, Podman uses containers without root privileges that run in isolated user namespaces. This method significantly reduces the risk of an outbreak. Docker is different: Container processes run as root by default. Because Podman containers are also tied to specific user sessions, the auditing systems can trace malicious activities back to specific users, which is much more difficult with Docker because of the global daemon.

However, the two platforms also share similarities: Podman and Docker draw on the capabilities of the Linux kernel and seccomp profiles to control process authorizations. Both environments can also configure robust security. That said, you are looking at significantly less overhead with Podman to achieve a comparable level of security.

systemd and Compose

Docker and Podman also differ in terms of container life cycle management. Because it lacks a daemon, Podman largely relies on systemd. Moreover, Podman can generate systemd service files from the running containers, which means you can let systemd manage your containers when you need to start, stop, and check on various services. Docker internally relies on the daemon to handle all of these tasks.

In terms of orchestration, Docker users generally rely on Docker Compose for local development because it makes it easy to define and manage multicontainer applications; Podman has a similar tool in the form of Podman Compose. Do not forget the pods themselves, which are based on an idea in Kubernetes that lets you manage a container group as a single unit. As far as deployment is concerned, Podman does not have an alternative to Docker Swarm; instead, it relies on external orchestration systems such as Kubernetes.

GUI and Dashboard

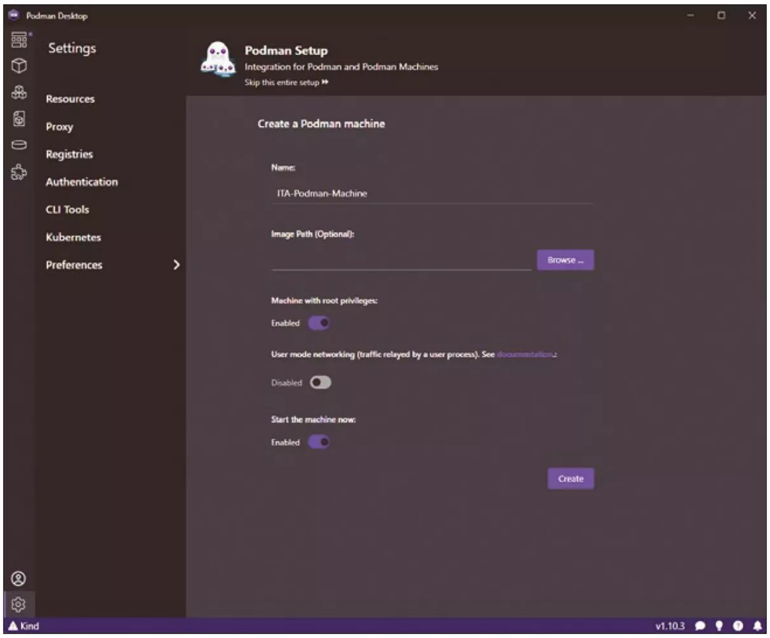
To simplify working with containers, the Red Hat development team announced the completion of Podman Desktop **[2]** in May 2023. Based on Docker Desktop, this environment also facilitates the handling of different types of containers. Podman Desktop also comes with a graphical interface to help you work seamlessly with containers and Kubernetes in a local environment, building a convenient bridge to Kubernetes environments.

The desktop environment comes with a clear-cut dashboard for interacting with and managing containers, images, and pods. You can also run Kubernetes objects locally with Podman or in local Kubernetes clusters, which allows for a seamless transition from development to deployment.

You also need to know that Podman and Docker are not necessarily mutually exclusive. On the contrary, Podman Desktop, for example, works with various container engines, including Docker. The developers describe Podman Desktop as a “multi-engine UI tool” **[3]**. More precisely, the tool is compatible with the Docker and Podman APIs, which means you can use all of the containers and images belonging to the different engines in parallel, making it easier to migrate from Docker to Podman as a useful side effect.

Getting Started

Podman Desktop is available for macOS, Linux, and Windows in the



Podman Setup

Integration for Podman and Podman Machines

Skip this entire setup »

Create a Podman machine

Name: ITA-Podman-Machine

Image Path (Optional):

Machine with root privileges: Enabled

User mode networking (traffic relayed by a user process). See documentation: Disabled

Start the machine now: Enabled

Figure 1: **Creating an initial Podman machine with the desktop environment.**

download area of the project website, with DMG packages for Mac systems, Flatpak packages for Linux computers, and an installer and a portable environment for Windows users. WinGet-based installation is also possible. If you opt for the installer version, the setup wizard offers to set up the Compose extension and the `kubectl` command-line interface in addition to Podman.

After installing Podman Desktop, you can set about creating your first Podman machine. On macOS and Windows, a Linux virtual machine is required to run the Podman container engine. If Podman is not already installed, you can go to the Resources menu and follow the *Setup* link. Podman Desktop checks whether Podman is installed. If not, press *Next*.

After installing Podman, you can immediately start creating your first Podman machine by assigning it a name in the *Create a Podman machine* box (Figure 1). Root privileges are assigned by default if the *Machine with root privileges* box is checked: You will want to change this setting. Now press *Create* to create the container. The *User mode networking* option lets you forward the guest's data traffic to the host by way of a userspace process. Setting this option to *true* ensures that virtual private networks evaluate all data traffic from the Podman system as originating from the host and avoids the risk of unwanted blocking.

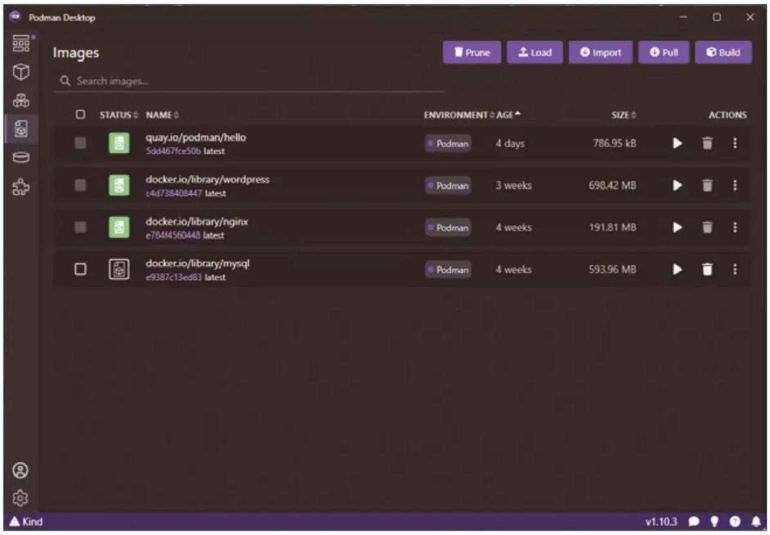
If you want to launch the machine immediately after creating it, choose the *Enabled* option. Finally, click *Create* to create the container. The new machine now appears on the dashboard. If you switch to the terminal window, you can use the

```
podman machine list
```

command to display a list of Podman machines.

To access containers, pods, images, and extensions, use the menubar on the left-hand side of the Podman Desktop window. In a fresh installation, all of these sections begin empty. To retrieve an initial image, go to the Images menu, specify the desired image, and press *Pull* (Figure 2). Podman Desktop displays a success message after the download. You will then find the new image in the Images menu. You can view the details of an image by clicking on the entry for the image. Besides a summary, Podman Desktop shows the history and details of the image in JSON format. You can also launch the image from the overview.

After pressing the *Start* button, the tool shows various tabs for configuring the container from the underlying image file. On the *Basic* tab, you can do such things as specify the access point and configure the port mappings and the container name. The settings in the *Networking* tab let you specify a hostname. Clicking on *Start Container* executes



☐	STATUS	NAME	ENVIRONMENT	AGE	SIZE	ACTIONS
☐		quay.io/podman/hello 5dd467fce50b latest	Podman	4 days	786.95 kB	
☐		docker.io/library/wordpress c4d738408447 latest	Podman	3 weeks	698.42 MB	
☐		docker.io/library/nginx e784f4560448 latest	Podman	4 weeks	191.81 MB	
☐		docker.io/library/mysql e9387c13ed83 latest	Podman	4 weeks	593.96 MB	

Figure 2: **After downloading various images, you can create a matching container with just a couple of mouse clicks.**

the configuration, which generates a matching entry in the Container menu, and you can view the logs on the *Logs* tab.

All of the container information, including information on the image you used, can be found on the *Inspect* tab. The *Kube* tab takes you to the Kubernetes YAML manifest file that the container uses. If you want to pass commands into the container, go to the *Terminal* tab to do so.

Managing Containers

What Podman Desktop does not tell you is which container images are available. To find out, you need to use a `podman search` term in the terminal window. If you want to set up an NGINX web server, for example, type the command,

```
podman search nginx
```

which searches for matches in all available repositories. In the resulting output, select the appropriate package and download it with the image manager `pull` function. The repositories are stored in the `/etc/containers/repositories.conf` configuration file.

Podman Desktop facilitates everyday use. Clicking on *Manage registries* shows which repositories are available. If you want to add your organization's own internal registries, select *Add registry*, enter the URL with the credentials, and press *Login* to access the repository.

Verifying that the registry setup is working properly is important. If you want to set up a Podman-based WordPress installation, for example, you would download the supplied image and generate a container based on this image. To enable access to the web-based environment, you have to specify the port mappings. After launching the environment, you access the blog system by selecting *Open Browser*.

Working with Images

To make migrating from Docker to Podman as simple as possible, the developers have implemented an import function that facilitates switching to or evaluating Podman.

To try it out, open the *Images* manager, click on the *Import* button, add the desired image, and launch the process by pressing *Import Containers*. As the button name suggests, you can import several images at once.

If you want to import an existing container, you need to save it first. In the case of a Docker container, run the command:

```
docker export <your_container> ➥
         <your_container_export>.tar
```

You can also use the import function to transfer a Podman container to a third-party system. To do so, first save the container with the command:

```
podman save <your_container> ➥
         <your_container_export>.tar
```

Things are even easier with Podman Desktop. You just need to run the `export` command in the container manager's Actions column. During the import process, make sure you only use lowercase letters for the image name; otherwise, Podman Desktop displays an error message. After importing, you will find the imported archive in the Podman image manager.

Docker Compose

Docker Compose allows you to generate multi-tier applications, such as those found a multitude of times in typical infrastructures. The orchestration tool lets you define different containers, which you can link to and run on any Docker host. The good news for Podman Desktop users is that the Podman environment also supports the Compose specification **[4]**.

Before you can use Compose in Podman Desktop, you need to install the Compose Engine. If you did not enable Compose during the Podman Desktop installation, do so now by going to the *Setting* | *Resources* menu item, clicking *Setup* in the Compose section, and following the instructions. The Compose reference implementation is located in your `PATH`, which explains why you can display the Compose version in the terminal window:

```
docker-compose version
```

To verify the installation, you want to make sure Podman has found the same Compose version:

```
podman compose version
```

To use Compose, you need a Compose file (e.g., `compose.yaml`); to execute it, access the terminal window:

```
podman compose --file compose.yaml up ➥
         --detach
```

Compose then starts the specified containers and services. The following labels are added to the various resources:

- Container name: `com.docker.compose.project`
- Service ID: `com.docker.compose.service`

Podman Desktop recognizes the Compose label and displays a matching icon to visualize the container group. A `(compose)` suffix is also added to the group name.

Advanced Techniques

Podman and Podman Desktop offer a number of other interesting features. For example, you can easily migrate Podman containers to Kubernetes after at least setting up a Kubernetes context. Podman Desktop automatically configures access to Kubernetes clusters when you create a child-based local Kubernetes cluster, generate a local OpenShift cluster, or configure access to a developer sandbox. If a Kubernetes cluster is available, you can view its details by right-clicking on the Podman Desktop icon. Use the *Kubernetes* submenu to identify the current context. This function is also used to deploy a container in a Kubernetes environment.

Another interesting feature of Podman Desktop is that you can use the environment to extend the functionality of the modular Podman system. You have the option of setting up extensions during the initial installation of the desktop environment. Extensions can, for example, enable support for virtual machines, such as Lima installations, and you can integrate third-party tools such as `kind` and the Docker Compose discussed here.

Available add-ons can be accessed from the Extensions menu (Figure 3). For example, you could set up *minikube* to use a local Kubernetes environment or *Podman AI Lab* to add large language model (LLM) support to Podman Desktop.

You can also configure the interaction with Red Hat OpenShift here. The Bootable Containers (bootc) option creates a bootable operating system from a container file. More specifically, you can convert containers into virtual machines or ISO and RAW images.

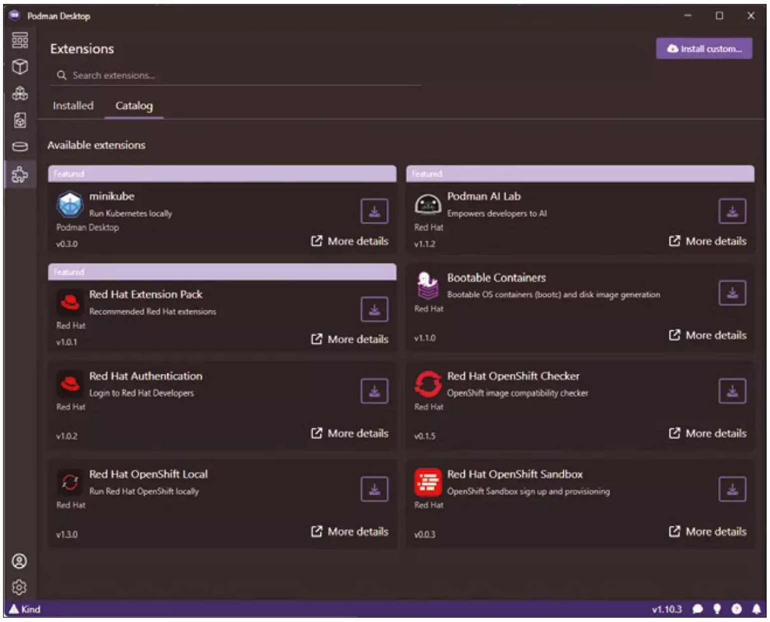

Podman Desktop
Extensions
Install custom...
Search extensions...
Installed
Catalog
Available extensions
Featured
minikube
Run Kubernetes locally
Podman Desktop
v0.3.0
More details
Featured
Podman AI Lab
Empowers developers to AI
Red Hat
v1.1.2
More details
Featured
Red Hat Extension Pack
Recommended Red Hat extensions
Red Hat
v1.0.1
More details
Bootable Containers
Bootable OS containers (bootc) and disk image generation
Red Hat
v1.1.0
More details
Red Hat Authentication
Login to Red Hat Developers
Red Hat
v1.0.2
More details
Red Hat OpenShift Checker
OpenShift image compatibility checker
Red Hat
v0.1.5
More details
Red Hat OpenShift Local
Run Red Hat OpenShift locally
Red Hat
v1.3.0
More details
Red Hat OpenShift Sandbox
OpenShift Sandbox sign up and provisioning
Red Hat
v0.0.3
More details
Kind
v1.10.3

Figure 3: The Extensions feature in Podman Desktop lets you extend your Podman environment.

Conclusions

Today, the assumption is that half of all corporations run around half of their business applications in container environments – and the numbers are rising. Against this backdrop, the use of tools such as Podman has long been business-critical. Podman Desktop makes a significant contribution by simplifying standard tasks with Podman. ■

Info

[1] Podman: [https://podman.io]

[2] Podman Desktop: [https://podman-desktop.io]

[3] Five facts for Docker users: [https://podman-desktop.io/blog/5-things-to-know-for-a-docker-user]

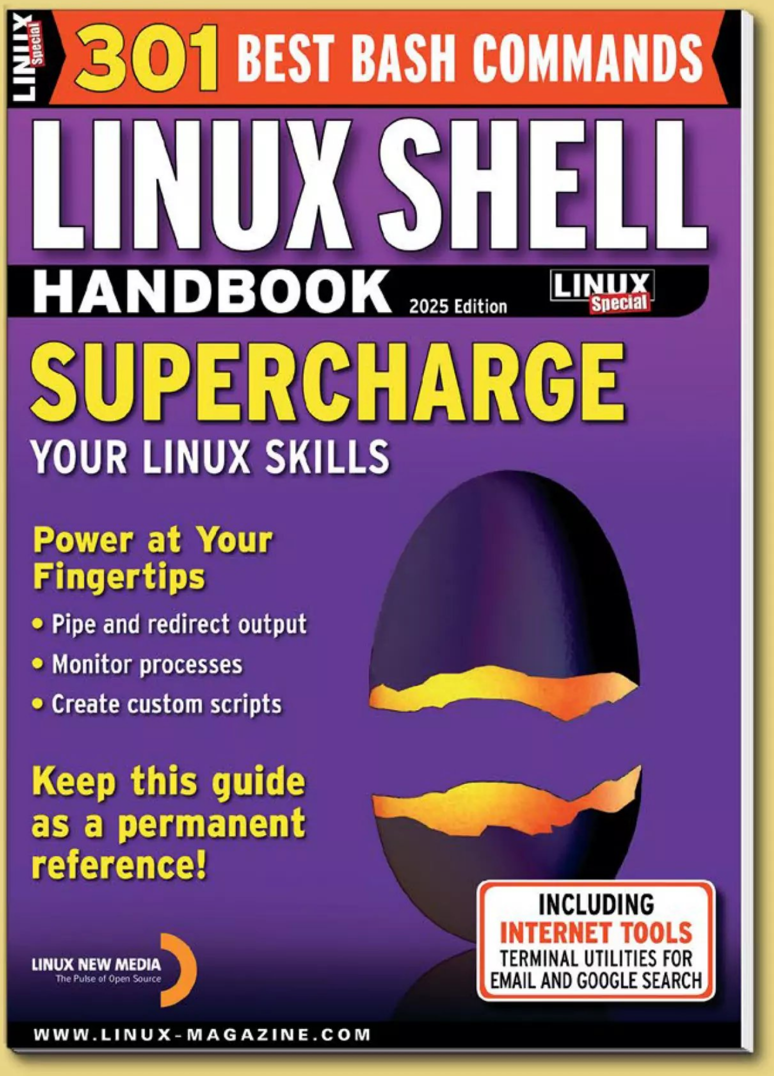
[4] Compose specification: [https://compose-spec.io]

The Author

Holger Reibold, computer scientist, has worked as an IT journalist since 1995. His main interests are open source tools and security topics.

THINK LIKE THE EXPERTS
Linux Shell Handbook 2025 Edition

This new edition is packed with the most important utilities for configuring and troubleshooting systems.



LINUX Special
301 BEST BASH COMMANDS
LINUX SHELL
HANDBOOK 2025 Edition
LINUX Special
SUPERCHARGE
YOUR LINUX SKILLS

Power at Your Fingertips
- Pipe and redirect output
- Monitor processes
- Create custom scripts

Keep this guide as a permanent reference!

INCLUDING
INTERNET TOOLS
TERMINAL UTILITIES FOR EMAIL AND GOOGLE SEARCH

LINUX NEW MEDIA
The Pulse of Open Source

WWW.LINUX-MAGAZINE.COM

Here's a look at some of what you'll find inside:

- Customizing Bash
- Regular Expressions
- Systemd
- Bash Scripting
- Networking Tools
- Internet Tools
- And much more!

ORDER ONLINE:
shop.linuxnewmedia.com/shop

Operating system images with bootc

Production Line

The bootc project lets you use a container-based workflow to create bootable operating system images. In a few simple steps, we show you how to use Podman to create ready-to-run images. By Thorsten Scherf

Container images generally only provide the components you need to run a specific application inside a container and do not include a Linux kernel, a bootloader, firmware, or similar components. Many workflows already in place create and manage container images for applications, typically with a zoo of tools that create an image, check it for vulnerabilities, and then publish it in a container store. This workflow would also be helpful for bootable system images – which is exactly where two projects, bootc [1] and bootc Image Builder [2], enter the scene.

Containers for Operating Systems

These bootc projects let you use the same workflow you have in place for application containers to create operating system containers. All you need is a containerfile or Dockerfile along with the bootc Image Builder service to create a bootable Open Container Initiative (OCI) image that you can start on a host or virtual machine.

An image created in this way also comes with the Linux kernel, which is loaded in the normal way at boot time. Therefore, the basic operating system does not reside in the container and a systemd process with a PID of 1 is available in the normal way. Transactional updates let you keep the system up to date. The updates create additional layers on the system on top of the base image. You just need to create an updated bootc base image, with no need for a new disk image. In this way, the update process is identical to other OSTree-based systems.

You can use the `bootc-image-builder` container tool to create bootable disk images with the aid of the Podman container runtime. This process generates a disk image from a base image. Alternatively, you can also use the graphical Podman Desktop [3] tool, which simplifies the container handling and comes with a selection of extensions – including some for bootc – to help create bootable container images. I will primarily be focusing on the bootc extension in this article.

Generating Images in a Terminal Window

To begin, I'll first take a look at the process of generating disk images in

Lead Image © alfazetchronicles, 123RF.com

a terminal window from a standard bootc image. Of course, you always have the options of extending this bootc image to include additional components or making other adjustments in advance with the use of a container file or Dockerfile. You then publish the customized image in a store, which is also necessary when, for example, updates become available for the image you are using. You can use either the RAW, QCOW2, VMDK, ISO, or Amazon Machine Image (AMI) formats for the disk image. In any case, you always need a bootc base image first. This special container image contains the kernel and the systemd service along with other components for a bootable system. Base images are available for Fedora, CentOS, and Red Hat Enterprise Linux systems. The following example shows that the components are included in the image:

```
# podman run --rm ↩
  -it quay.io/centos-bootc/centos-bootc:↩
   stream9 rpm ↩
  -q kernel systemd podman bootc
kernel-5.14.0-467.el9.aarch64
systemd-252-37.el9.aarch64
podman-5.1.0-1.el9.aarch64
bootc-0.1.11-2.el9.aarch64
```

Because this kind of image does not contain a user account by default, you can simply add one by editing the configuration file in TOML format and assign an SSH key and a group directly to the user:

```
[[customizations.user]]
name = "tscherf"
password = "Secret23"
key = "ssh-rsa AAA ... tscherf@domain.com"
groups = ["wheel"]
```

In this example, I used `centos-bootc:stream9` as the bootc base image and passed it into the `bootc-image-builder` service. The completed disk image uses the QCOW2 format. If everything goes well, the new image will be available in the output directory – but don't hold your breath. You can then launch the build process for the CentOS Stream 9 image itself, as in Listing 1.

Booting the Image

To boot the disk image you just created for test purposes, you can use a virtualization tool such as libvirt. This example,

```
# virt-install ↩
  --name bootc ↩
  --memory 4096 ↩
  --vcpus 2 ↩
  --disk qcow2/disk.qcow2 ↩
  --import
```

uses `virt-install`, which is based on libvirt.

Podman Desktop

The Podman Desktop graphical front end is an alternative to the build process in a terminal. The tool provides a bootc extension and then automatically takes care of generating a new bootable container image for you. To do so, press the *Build* button and then simply select the desired bootc base image from the pull-down menu along with the desired image format. The tool then automatically launches the Image Builder Service in a container and creates the new image there.
Of course, you can also adapt the bootc base image to your own requirements up front before you create a bootable disk image from it. This example,

```
FROM quay.io/centos-bootc/centos-bootc:↩
     stream9
RUN dnf -y install httpd && ↩
    systemctl enable httpd && ↩
    mv /var/www /usr/share/www && ↩
    sed -ie 's,/var/www,/usr/share/www,' ↩
      /etc/httpd/conf/httpd.conf
RUN rm -rf /usr/share/httpd/noindex
COPY index.html /usr/share/www/html
EXPOSE 80
```

sets up a simple container file for running a web server in a bootc container.

Conclusions

Container images for apps are nothing new, but what about bootable operating systems? Armed with bootc container images and a little help from the bootc Image Builder service, you can quickly create bootable container images for your choice of operating system. You can then use your container-based workflow for both your applications and your operating systems. You will find more information about this new deployment method for operating systems on Red Hat's image mode page [4]. ■

Info

[1] bootc: [https://github.com/containers/bootc]

[2] bootc Image Builder: [https://github.com/osbuild/bootc-image-builder]

[3] Podman Desktop: [https://podman-desktop.io]

[4] RHEL image mode: [https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux/image-mode]

The Author

Thorsten Scherf is the global Product Lead for Identity Management and Platform Security in Red Hat's Product Operations group. He is a regular speaker at various international conferences and writes a lot about open source software.

Listing 1: Build Process

```
# podman run --rm -it --privileged --pull=newer --security-opt label=type:unconfined_t -v $(pwd)/config.toml:/config.toml -v $(pwd)/output:/output -v
  /var/lib/containers/storage:/var/lib/containers/storage quay.io/centos-bootc/bootc-image-builder:latest --type qcow2 --local quay.io/centos-bootc/
  centos-bootc:stream9
```

Mobile hacking tool

Small but Deadly

The Flipper Zero mobile hacking tool, a highly portable device that looks like a kiddy toy at first glance, can check a wide range of wireless connections for vulnerabilities. We take a look at potential applications in pentesting scenarios.

By Holger Reibold

The story of Flipper Zero began in 2019 when Alex Kulagin and Pavel Zhovner set about raising funds for their business idea on Kickstarter. The rest is more or less history. Ever since Flipper Zero went viral on Tiktok as a "Tamagotchi for hackers," there was no holding it back. The media have repeatedly associated Flipper Zero with the same potential applications: replicating electronic cash (EC) or credit card data and unlocking cars and hotel rooms. Smartphones can also be unlocked in a matter of seconds. Of course, these are just the applications that make a splash in the media.

These capabilities have also given rise to criticism. In Canada, the device was banned for a time because of its capabilities. The government has since lifted the ban, stating that its use is only permitted by "legitimate actors." The reason for the ban was the possibility of discovering security vulnerabilities in vehicles. The Flipper manufacturer came out on top, however, with the support of the "stop the absurd ban" petition at Change.org [1], arguing that government institutions should focus on forcing vehicle car manufacturers to eliminate security flaws rather than banning the tool that detects them. Amazon had also temporarily stopped selling Flipper Zero. In Brazil, the devices were even confiscated on the grounds that their wireless interfaces did not have official clearance.

None of this fuss has done Flipper Zero any harm; on the contrary, these seemingly helpless attempts to ban hacking tools have helped the Flipper developers attract massive media attention. At the same time, state actors often overlook the fact that even beginners can learn hacking techniques relatively easily with freely available tools such as Kali Linux.

Capabilities

Wireless communications are both a blessing and a curse. Even if basic security is usually in place, the door is often left open to potential attackers – especially when admins fail to adhere to the latest standards and fail to update software. After the successful conclusion of the Kickstarter campaign in 2020, which raised the impressive sum of $4.8 million according to the Hackaday platform [2], an active community has sprung up around this practical device, helping to drive the project forward. From the media coverage, you might be excused for gaining the impression that Flipper Zero is a master key for Internet-of-Things services, but this is more of a media exaggeration. Flipper Zero has a set of antennas suitable for capturing, storing, cloning, and emulating radio signals. More specifically, the tool supports the following signal types:

Photo by Valentina Curini on Unsplash

- Near-field communication (NFC): Allows NFC signals to be read from bank cards and building access cards.
- 125kHz radio-frequency identification (RFID): Older cards and microchips use this frequency.
- Infrared: Many remote controls use infrared signals.
- Sub-1GHz: Garage door remote controls and keyless remote control systems in particular use sub-1GHz frequencies for communication.

The CC1101 transceiver chip is the beating heart of the Flipper Zero; it supports recording, analyzing, and replaying RF signals from 300 to 928MHz. The user can leverage this ability to hijack remote control systems that do not use a rolling code for authentication and are therefore susceptible to replay attacks. Moreover, the Flipper Zero can read and emulate various RFID cards, record IR signals, or execute transmitted and programmable user data.

The device also acts as a USB-to-UART/SPI/I2C adapter and supports Bluetooth spamming. This low-energy denial-of-service (DoS) attack can be used to crash iPhones and iPads in particular. Devices with iOS 17.2 or newer are less susceptible to this type of attack. Last but not least, Flipper Zero can act as a (wireless) BadUSB, which is an exploit for manipulating the firmware of USB devices.

Flipper uses the FreeRTOS real-time operating system and can also be used as a human interface device (HID) controller to replace, say, PC keyboards or mice. The modular architecture is another highlight, allowing the expansion of the basic system. The WiFi-capable developer board lets you add debugging and firmware update functions to the device.

The board (see the "Modules and Prices" box) is based on the ESP32-S2 microcontroller unit (MCU) with custom firmware that includes Black Magic Debug and CMSIS-DAP – a protocol and interface for programming and debugging microcontrollers – and is built on the ESP-IDF, Espressif's official development framework for the ESP32 series of microcontrollers providing libraries, tools, and sample code for developing applications on the ESP32 MCU. This module lets you flash and debug the Flipper Zero over WiFi or USB cable. The developer board can also act as a USB-to-UART adapter. The UART hardware component supports serial communications and converts data between parallel and serial formats.

Flipper Zero can also be extended to include a video game module on a Raspberry Pi RP2040 microcontroller and opens up additional entertainment and development options. Like the developer board, the module is connected to Flipper Zero by GPIO connections. Even though the Flipper is often referred to as the "Swiss army knife" of penetration testing (pentesting), it is clear in practice that the device, or more precisely its antennas, support a very limited frequency spectrum; the software features are also limited. Software-defined radio (SDR) opens up significantly more possibilities than those used by Flipper Zero.

Reading Out NFC and Testing BadUSB

Flipper Zero is primarily associated with its ability to clone credit cards, open modern vehicles, and break digital security systems. The device has a read mode for typical remote control functions. To read an NFC chip, position the card behind the device, open *NFC* tab (**Figure 1**), and execute the *Read* command. After reading, Flipper presents the result, which you can save and use with the *Emulate* function to, for example, verify the reliability of blocking cards or sleeves, which are designed to prevent NFC and RFID skimming. BadUSB has become a popular attack vector, which explains why you will want to test your IT security's resilience to it. In this mode, the computer regards the Flipper Zero device as a HID. Flipper uses DuckyScript **[4]** to execute commands on the target system. The BadUSB app requires an installed microSD card, because Flipper Zero stores the data there. That aside, the BadUSB function is easy to use. Starting at the Main Menu, choose *Bad USB*, select the payload script, connect the device to the laptop, and *Run* the script. Flipper Zero has a number of sample scripts that you can use to gain some initial experience. You will find another BadUSB example in the *Apps | Scripts* menu. The two most comprehensive collections of scripts (e.g., which you can use to read passwords in the scope of a pentest) are available for download from GitHub **[5] [6]**.

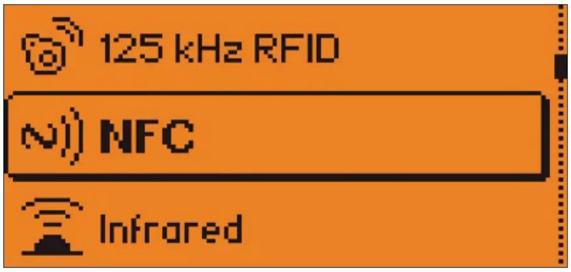
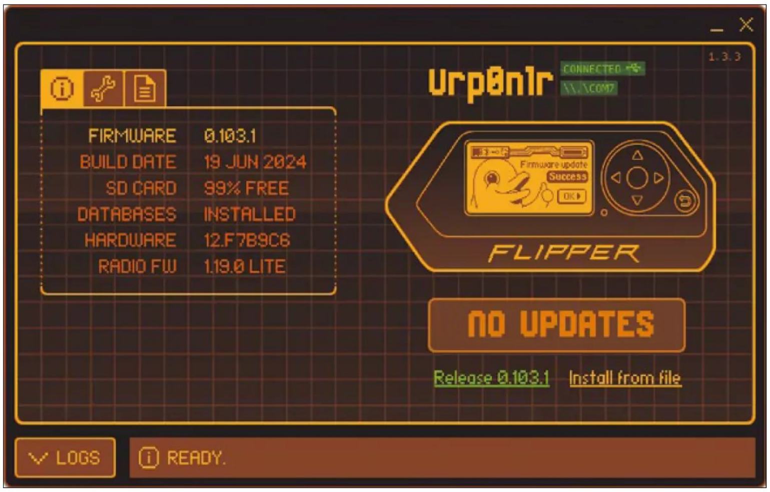


Figure 1: From the main menu, you can navigate to the various function areas.

Modules and Prices

The developers sell the device, which is made in China, from their online store **[3]**. The price for the basic device is $169 (EUR199). Purchasing a silicone case ($15, EUR18) is also recommended for more intensive use. The developer board will set you back $29 (EUR34) and the video game module $49 (EUR59). The developers also offer prototyping boards that you can expand with your own functions.

Flipper Zero is designed for interaction with a smartphone. During the initial start-up, for example, you need to update the firmware, which you also need to do if you want to use the developer board. From the Flipper app, you can download the update from the App Store or Google Play. The developers also provide qFlipper for macOS, Linux, and Windows in their download area **[7]**. This app can



FIRMWARE 0.103.1
BUILD DATE 19 JUN 2024
SD CARD 99% FREE
DATABASES INSTALLED
HARDWARE 12.F7B9C6
RADIO FW 1.19.0 LITE

Figure 2: **qFlipper is very useful for installing updates or firmware and is ideal as a remote control.**

install updates and control Flipper Zero, as well (Figure 2). Also on offer is an application SDK for developing your own Flipper Zero applications in the form of the micro Flipper build tool (µFBT).

Testing WiFi Security

WiFi network reliability and security is probably high on the agenda for most admins. Flipper Zero can offer valuable services here, but you need both the developer board and to install various add-on components. The plugin board's design is unfortunate: Even a lack of attention can cause damage. Resourceful users might want to use a 3D printer to create a case, and others can pick one up from well-known online sources. Note that the two buttons on the circuit board are for rebooting and resetting.

To use the WiFi adapter to test a wireless environment, you need the Marauder module [8]. Start by unpacking the ZIP archive after downloading.

Then things get a little tricky: Press the *Boot* button on the board and connect the developer board to the USB-C cable of a computer, wait three seconds, then release the button.

Run the `flash.bat` batch file to pop up the Marauder Flasher script and select option *1. Flash Marauder (no SD mod) to Devboard*. You can follow the flashing process on the console. When done, close the console, disconnect the board from the computer, and connect it to your Flipper Zero. You do not need to restart. Once plugged in, the board lights up blue, green, and red in sequence, which indicates that the setup process was error free. From now on, the Marauder module is available under *App* | *GPIO*.

Besides the standard firmware, you can install other alternatives to open various additional functions. Some of the more interesting alternatives include Unleashed Firmware [9] and Xtreme Firmware [10]. Both expand the feature set in a useful way; for example, you can add menus, modify animations, and define keyboard shortcuts.

Unleashed also lets you rename the device – an option you do not get with most alternative firmware. The nice thing about these two variants is that they offer a web-based installer. To install, simply browse to the URL and make sure that Flipper Zero is connected to a computer with an Internet connection. The web installer takes care of everything else. The only restriction is that it is only supported by Chrome and Opera. After the download, you can follow the installation process on the Flipper display.

One advantage of the Xtreme firmware is not only a simplified menu navigation but a plethora of additional features that are not included with the standard firmware. The Xtreme firmware proves to be practical for pentesting WiFi networks by creating an *Apps* | *WiFi* submenu. Besides Marauder, you will also find the Wardrive and Deauther modules and a WiFi scanner here.

Marauder is easy to use: The *Scan* item checks the environment for available access points. WiFi station mode is also supported. To attack a specific access point, switch to the *List* item and specify the number Marauder assigns to the target. Next, go to *Select*, specify the target with the `select -a <target number>` command, and tap *Save* to save the selection.

You now have the choice of attacking the access point directly from *Attack* or recording the data traffic with the integrated sniffer. Flipper saves the recorded data on the microSD card. You can analyze the PCAP file with Wireshark [11] and extract WiFi passwords with hashcat [12].

Given the small size of the housing – Flipper Zero is just 10 cm wide and 4 cm high – it is not surprising that the performance of the integrated antennas is limited. If you want to test wireless connections intensively, you might want to consider purchasing an additional Flipper antenna to improve the transmission and reception performance.

Becoming Flipper One

Four years have passed since the release of Flipper Zero – an eternity in the fast-moving IT industry. The name "Flipper Zero" suggests that the next version could be named "Flipper One," and indeed, the Flipper developers announced the upcoming version last spring before the announcement disappeared from the company website virtually overnight – although it still exists in the Internet Archive (Flipper Devices 2023). Many signs indicate that the developers are pushing ahead with further development, including improved hardware and Kali Linux support, and the community is eagerly awaiting the successor to the Zero version, which will – that much seems certain – be called Flipper One.

Initial reactions from the Flipper community to the One announcement were positive, according to which, it will be based on an i.MX6 SoC and feature open source hardware and software. The system-on-chip (SoC) will have a 1GHz CPU and integrate WiFi support. The improved NFC module should be able to read and write all ISO 14443 cards.

According to what is known thus far, the developers have a combination of field-programmable gate array (FPGA) and SDR in mind. The question of the best choice of WiFi module has also proved problematic because many chips are outdated. Also unclear is whether architectural considerations are responsible for the delays. Costing seems to be having an influence on the launch date in view of the extended features. Given the unclear future of Flipper, it can be assumed that it will be some time before Flipper One is released.

Conclusions

Flipper Zero is a practical multifunctional tool that allows you to discover vulnerabilities in your infrastructures and system configurations. It can be used to check a variety of devices and services for potential configuration errors. The greatest benefit is probably that you can go walkabout and show people in real time how easily vulnerabilities can be exploited by attackers.

Info

[1] Flipper Zero ban in Canada: [https://www.change.org/p/stop-the-absurd-ban-of-flipper-zero-in-canada]
[2] Funding: [https://hackaday.com/2020/09/02/flipper-zero-blasts-past-funding-goal-and-into-our-hearts/]
[3] Flipper Zero online store: [https://shop.flipperzero.one]
[4] DuckyScript: [https://docs.hak5.org/hak5-usb-rubber-ducky]
[5] BadUSB by CharlesTheGreat77: [https://github.com/CharlesTheGreat77/BADUSB]
[6] BadUSB by nocomp: [https://github.com/nocomp/Flipper_Zero_Badusb_hack5_payloads]
[7] Downloads: [https://flipperzero.one/downloads]
[8] Marauder module: [https://github.com/justcallmekoko/ESP32Marauder/wiki/flipper-zero]
[9] Unleashed firmware: [https://github.com/DarkFlippers/unleashed-firmware]
[10] Xtreme firmware: [https://github.com/Flipper-XFW/Xtreme-Firmware]
[11] Wireshark: [https://www.wireshark.org]
[12] hashcat: [https://hashcat.net/hashcat/]

The Author

Holger Reibold, computer scientist, has worked as an IT journalist since 1995. His main interests are open source tools and security topics.

Shop the Shop shop.linuxnewmedia.com/shop

Missed an issue?

You're in luck.

Most back issues are still available. Order now before they're gone!

shop.linuxnewmedia.com/shop

GET IT NOW!

SAVE TIME ON DELIVERY WITH OUR ALTERNATIVE PDF EDITIONS

Identify malicious traffic with Maltrail

Tracker

This traffic analysis software identifies malicious traffic on your network with the use of established sources.

By Holger Reibold

As digitalization makes further headway into business processes, the vulnerability of IT infrastructures increases. Just one undetected vulnerability could open the door to potential attackers. Even if 100 percent protection remains an illusion, organizations need to be proactive in the face of an increasingly stressful security landscape. Maltrail is a free tool that can be a useful addition in your search for would-be attackers on your network.

Detection

Managing legacy intrusion detection and intrusion prevention systems involves a great deal of effort, partly because hybrid approaches consisting of intrusion detection (IDS), host-based intrusion detection (HIDS), and network-based intrusion detection (NIDS) systems are required. However, managing Maltrail is child's play.

To identify malicious traffic on your network, Maltrail uses publicly available lists that point out critical or suspicious traces. On the GitHub project site [1], the developers reveal the lists and malicious entities on which they draw. Maltrail pays particular attention to statistics, antivirus (AV) reports, domain names, and IP addresses. HTTP user agent header values – such as those generated by `sqlmap` [2], used for automatic SQL injection and database hijacking – are also considered. You can also opt to deploy heuristic mechanisms, which can be useful in detecting previously unknown threats. The various suspicious sources are jointly referred to as "trails."

Basic Structure

Maltrail's strengths lie in its ability to detect suspicious web access and name resolution events. The functionality is largely limited to analysis and report output. The environment features a sensor and a server component. The server is used to store events and provide web-based access, whereas the sensor scrutinizes data traffic. The server in the Python-based environment is implemented in the `server.py` file and the sensor, unsurprisingly, in the `sensor.py` file.

In the default configuration, the server and sensor run on the same system. The developers claim that a virtually arbitrary number of events can be analyzed and processed thanks to the use of compression techniques. In principle, you can also use the sensor as a standalone environment. In this case, all the events are stored in the local logging directory, and the log entries can be evaluated manually or with the help of a third-party application that offers a CSV import function.

If the sensor registers a positive match, it sends the detailed information to the Maltrail server, where it is stored in the log directory. In a default configuration, in which both Maltrail components run on the same computer, the tool writes the logs to the local log directory, as mentioned previously. If you prefer to run the two components separately, you need to edit the `LOG_SERVER` configuration. In this case, the sensor sends messages by the User Datagram Protocol (UDP) to the remote server. For an initial impression of Maltrail, the developers offer an online demo [3].

Maltrail is not particularly demanding in terms of resource requirements. In addition to a Python interpreter (version 2.6 or greater), you need to install the *pcapy-ng* package. If you use Python 3.x, this will be *pcapy* instead. You need to allow at least 1GB of RAM

Lead Image © Andrea De Martin, 123RF.com

in single-process mode for the sensor component, which also requires root privileges. The server component has no specific requirements, and the install process is simple. If you are working with Ubuntu or Debian, just run the following commands:

```
sudo apt-get install ➥
  git python3 python3-dev python3-pip ➥
  python-is-python3 libpcap-dev ➥
  build-essential procps schedtool
sudo pip3 install pcapy-ng
git clone --depth 1 ➥
  https://github.com/stamparm/maltrail.git
cd maltrail
sudo python3 sensor.py
```

Alternatively, you can use a Docker container. To start the server on the same system, open a new terminal window and type the commands:

```
[[-d maltrail]] || git clone --depth 1 ➥
  https://github.com/stamparm/maltrail.git
cd maltrail
python server.py
```

When the sensor is started for the first time, it downloads the available block and reputation lists to the local system, which you can monitor easily in a terminal window. After launching the server, you can access Maltrail in the web interface, which for a local installation is *http://127.0.0.1:8338* with default access credentials *admin: changeme!*

Initial Configuration Steps

The Maltrail environment is configured in the `maltrail.conf` file, which has `[Server]` and `[Sensor]` sections to let you configure both components. Specify the server address and port in `HTTP_ADDRESS` and `HTTP_PORT`. If you want secure access by SSL or TLS, set the `USE_SSL` option to `true`. For secure access to work, the `SSL_PEM` option must refer to the certificate container.

Once you have had an initial look around, you will want to change the admin user. If needed, you can create more users. The settings for this can be found in the `USERS` subsection. Each user entry consists of the following data:

```
Username:sha256(Password):UID➥
        :filter_netmask
```

The UID value assigns a unique ID to the user. It is advisable to assign values `1000` or below to administrative accounts; for example:

```
admin:g6j0pelqmhg2n72zx9471f72rxf4iqzzml➥
      9zzrcnfzgreguqx69nopme6ctq8r8s➥
     :1000:192.168.0.0/16
```

The `UDP_ADDRESS` and `UDP_PORT` options specify the server that receives the messages. No changes are required for a default configuration.

When the server is started for the first time with

```
python server.py
```

the enabled `USE_SERVER_UPDATE_TRAILS` option makes sure the trails are updated from the trail definitions to guarantee that log entries end up in the logging directory and the reporting interface is available.

The sensor also offers various configuration options. If you are monitoring larger environments, you will want to give Maltrail additional resources. The easiest way to do this is to set the `USE_MULTIPROCESSING` option to `true`, which means that Maltrail will use all the CPU cores, with one core for the environment and all the others for packet processing. If you do not use this option, all tasks are handled by a single core.

The value of the information returned by Maltrail largely depends on the trails that the environment picks up from various sources. The `UPDATE_PERIOD` option determines the update interval in seconds. The default value of `86,400` ensures a daily update.

Besides trail-based analysis, Maltrail uses heuristics for risk assessment, which is enabled by default with `USE_HEURISTICS`. The check examines the traffic specifically for suspiciously long domain names or attempts to download EXE files. The disadvantage here is that heuristics can lead to a noticeable increase in false positives.

You can use the `SYSLOG_SERVER` option to forward the events to an external environment. The event data is sent to your choice of UDP service in the common event format (CEF), which includes Syslog, for example. If you prefer JSON as the format on the receiving end, use the `LOGSTASH_SERVER` option instead. Again, UDP is the communication protocol.

Finally, Maltrail has cross-component settings in the `[All]` section, which you use to decide whether debug messages are output globally in the environment or whether you want to use a proxy server. You can also change the default storage location for the trails here.

Informative Reporting

After logging in to the web interface (Figure 1), Maltrail comes up with a summary and a table of the logged events. The summary lists the number of threats, events, sources, and trails used. The environment also provides threat severity assessments. Clicking the *Events* button opens the events logged in the selected time period; the environment generates a diagram of the event history for the past few hours. The blue line shows DNS-based events, the yellow graph URL-specific events, and the red graph IP-specific events.

If you want to find out which sources are primarily responsible for the events, follow the *Sources* link. A stacked bar chart visualizes the major sources. Mousing over a column shows the event distribution by protocol. The *Trails* overview visualizes the share of critical trails as a pie chart. You can also call up details here by mousing over the views.

The tabular event overview has a wealth of detailed information on the various incidents, with color highlighting in the severity column helping to distinguish critical from less critical incidents. For each entry, you can call up the source IP address, the port, the protocol type, the associated trail, and the event type. The sort function is useful, for example, to find critical entries first.

The table view is again interactive. If you mouse over the scr_ip column, Maltrail displays the event details, and you can use the search function to narrow down the event list. Typing *high* restricts the view to events with that severity level.

In practice, the question arises as to how to identify specific processes. The Tor browser is popular with hackers, so if Maltrail detects an incident in which Tor was used, the program notes this in the info column and flags it as *tor*. You can restrict the table view to the events in which you are interested by typing an appropriate entry. Malware can also be identified in this way.

Limitations

Maltrail focuses on identifying and evaluating malicious data traffic, as shown; however, Maltrail does have some limitations. For a complete overview of relevant events, integration with an environment such as the Malware Information Sharing Platform (MISP) **[4]** might be a good idea, but it has no smart solution as yet.

The integration of various sensors into the web GUI has also proved difficult to date. Unfortunately, Maltrail does not offer any meaningful support for dealing with false positives, so admin experience is required. Criticism on some forums also claim that Maltrail quickly collects many gigabytes of data, depending on the size of the network.

One criticism – that Maltrail only plays a passive role on various occasions – is not fair, because interaction with more active tools is easy to implement. The Maltrail server configuration contains a `FAIL2BAN_REGEX` option, which allows sensor data to be transmitted to ipset or iptables. More specifically, this option enables the use of IP-based blocking mechanisms by regularly retrieving the relevant data. For example, you can integrate Maltrail into your security architecture with a small script that runs once a minute as a root cron job:

```
#!/bin/bash
ipset -q flush maltrail
ipset -q create maltrail hash:net
for ip in ➥
  $(curl http://127.0.0.1:8338/fail2ban ➥
    2>/dev/null | grep -P '^[0-9.]+$'); ➥
  do ipset add maltrail $ip; done
iptables -I INPUT -m set ➥
         --match-set maltrail src -j DROP
```

Maltrail is currently at version 0.68, but it has no roadmap. When I asked what features admins can expect in the future, I did not receive an answer.

Conclusions

"Small but powerful" is an apt description of Maltrail. The analysis software promises security-specific added value through traffic analysis with the use of established sources. ■

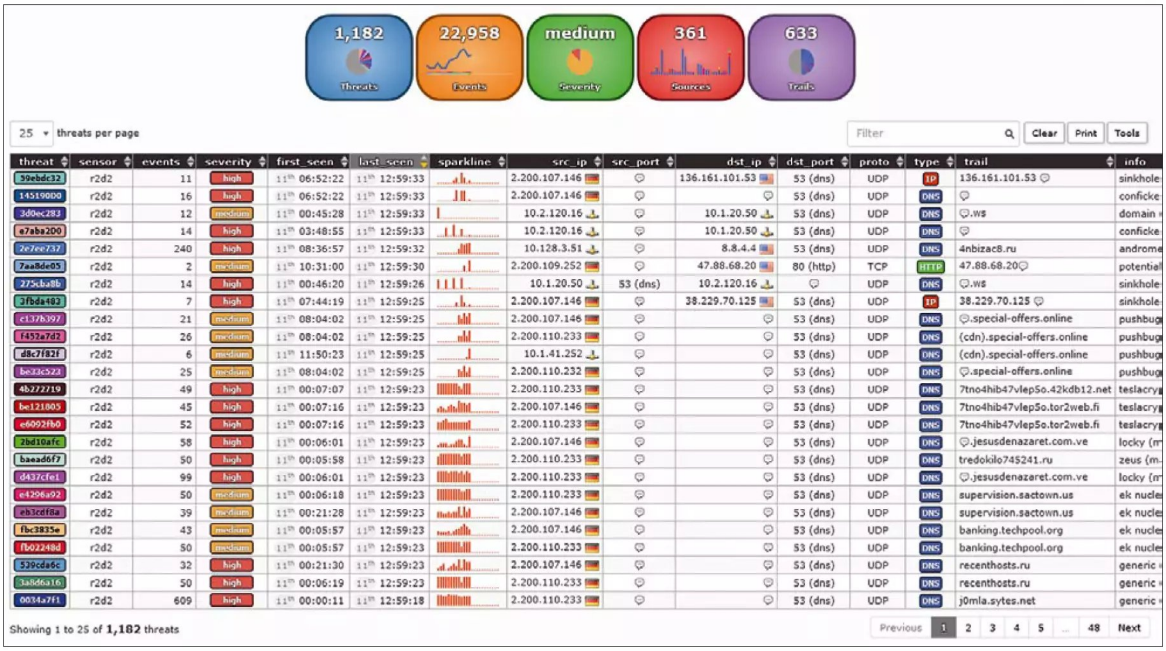
Info

[1] Maltrail: [https://github.com/stamparm/maltrail/]

[2] sqlmap: [https://sqlmap.org]

[3] Maltrail online demo: [https://maltraildemo.github.io]

[4] MISP threat sharing: [https://www.misp-project.org]



threat	sensor	events	severity	first_seen	last_seen	sparkline	src_ip	src_port	dst_ip	dst_port	proto	type	trail	info
59ebdc32	r2d2	11	high	11th 06:52:22	11th 12:59:33		2.200.107.146		136.161.101.53	53 (dns)	UDP	IP	136.161.101.53	sinkhole
14519000	r2d2	16	high	11th 06:52:22	11th 12:59:33		2.200.107.146			53 (dns)	UDP	DNS		conficke
3d0ec283	r2d2	12	medium	11th 00:45:28	11th 12:59:33		10.2.120.16		10.1.20.50	53 (dns)	UDP	DNS	.ws	domain
e7aba200	r2d2	14	high	11th 03:48:55	11th 12:59:33		10.2.120.16		10.1.20.50	53 (dns)	UDP	DNS		conficke
2e7ee737	r2d2	240	high	11th 08:36:57	11th 12:59:32		10.128.3.51		8.8.4.4	53 (dns)	UDP	DNS	4nbizac8.ru	androme
7aa8de05	r2d2	2	medium	11th 10:31:00	11th 12:59:30		2.200.109.252		47.88.68.20	80 (http)	TCP	HTTP	47.88.68.20	potential
275cba8b	r2d2	14	high	11th 00:46:20	11th 12:59:26		10.1.20.50	53 (dns)	10.2.120.16		UDP	DNS	.ws	sinkhole
3fbda483	r2d2	7	high	11th 07:44:19	11th 12:59:25		2.200.107.146		38.229.70.125	53 (dns)	UDP	IP	38.229.70.125	sinkhole
c137b397	r2d2	21	medium	11th 08:04:02	11th 12:59:25		2.200.107.146			53 (dns)	UDP	DNS	.special-offers.online	pushbug
f452a7d2	r2d2	26	medium	11th 08:04:02	11th 12:59:25		2.200.110.233			53 (dns)	UDP	DNS	(cdn).special-offers.online	pushbug
d8c7f82f	r2d2	6	medium	11th 11:50:23	11th 12:59:25		10.1.41.252			53 (dns)	UDP	DNS	(cdn).special-offers.online	pushbug
be33c523	r2d2	25	medium	11th 08:04:02	11th 12:59:25		2.200.110.232			53 (dns)	UDP	DNS	.special-offers.online	pushbug
4b272719	r2d2	49	high	11th 00:07:07	11th 12:59:23		2.200.110.233			53 (dns)	UDP	DNS	7tno4hib47vlep5o.42kdb12.net	teslacry
be121805	r2d2	45	high	11th 00:07:16	11th 12:59:23		2.200.107.146			53 (dns)	UDP	DNS	7tno4hib47vlep5o.tor2web.fi	teslacry
e6092fb0	r2d2	52	high	11th 00:07:16	11th 12:59:23		2.200.110.233			53 (dns)	UDP	DNS	7tno4hib47vlep5o.tor2web.fi	teslacry
2bd10afc	r2d2	58	high	11th 00:06:01	11th 12:59:23		2.200.107.146			53 (dns)	UDP	DNS	.jesusdenazaret.com.ve	locky (m
baead6f7	r2d2	50	high	11th 00:05:58	11th 12:59:23		2.200.110.233			53 (dns)	UDP	DNS	tredokilo745241.ru	zeus (m
d437cfe1	r2d2	99	high	11th 00:06:01	11th 12:59:23		2.200.110.233			53 (dns)	UDP	DNS	.jesusdenazaret.com.ve	locky (m
e4296a92	r2d2	50	medium	11th 00:06:18	11th 12:59:23		2.200.110.233			53 (dns)	UDP	DNS	supervision.sactown.us	ek nucle
eb3cdf8a	r2d2	39	medium	11th 00:21:28	11th 12:59:23		2.200.107.146			53 (dns)	UDP	DNS	supervision.sactown.us	ek nucle
fbc3835e	r2d2	43	medium	11th 00:05:57	11th 12:59:23		2.200.107.146			53 (dns)	UDP	DNS	banking.techpool.org	ek nucle
fb02248d	r2d2	50	medium	11th 00:05:57	11th 12:59:23		2.200.110.233			53 (dns)	UDP	DNS	banking.techpool.org	ek nucle
539cda6c	r2d2	32	high	11th 00:21:30	11th 12:59:23		2.200.107.146			53 (dns)	UDP	DNS	recenthosts.ru	generic
3a8d6a16	r2d2	50	high	11th 00:06:19	11th 12:59:23		2.200.110.233			53 (dns)	UDP	DNS	recenthosts.ru	generic
0034a7f1	r2d2	609	high	11th 00:00:11	11th 12:59:18		2.200.110.233			53 (dns)	UDP	DNS	j0mla.sytes.net	generic

Showing 1 to 25 of **1,182** threats

Figure 1: The Maltrail web GUI is fairly plain. The filter feature lets you restrict the view to events in which you are interested.

■

Linux Magazine Subscription

Print and digital options
12 issues per year

LibreOffice Alternatives
Optimizing Linux
What Comes After Git?

SUBSCRIBE
shop.linuxnewmedia.com/shop

Expand your Linux skills:

- In-depth articles on trending topics, including Bitcoin, ransomware, cloud computing, and more!
- How-tos and tutorials on useful tools that will save you time and protect your data
- Troubleshooting and optimization tips
- Insightful news on crucial developments in the world of open source
- Cool projects for Raspberry Pi, Arduino, and other maker-board systems

Go farther and do more with Linux, subscribe today and never miss another issue!

Follow us

@linux-magazine.com
Linux Magazine
@linuxpromagazine
@linuxmagazine@fosstodon.org

Need more Linux?

Subscribe free to Linux Update

Our free Linux Update newsletter delivers insightful articles and tech tips to your inbox every week.

https://bit.ly/Linux-Update

LINUX UPDATE

No. 397 • February 6, 2025

HETZNER OBJECT STORAGE BUILT TO GROW WITH YOUR NEEDS

Bringing Privacy to Social Media

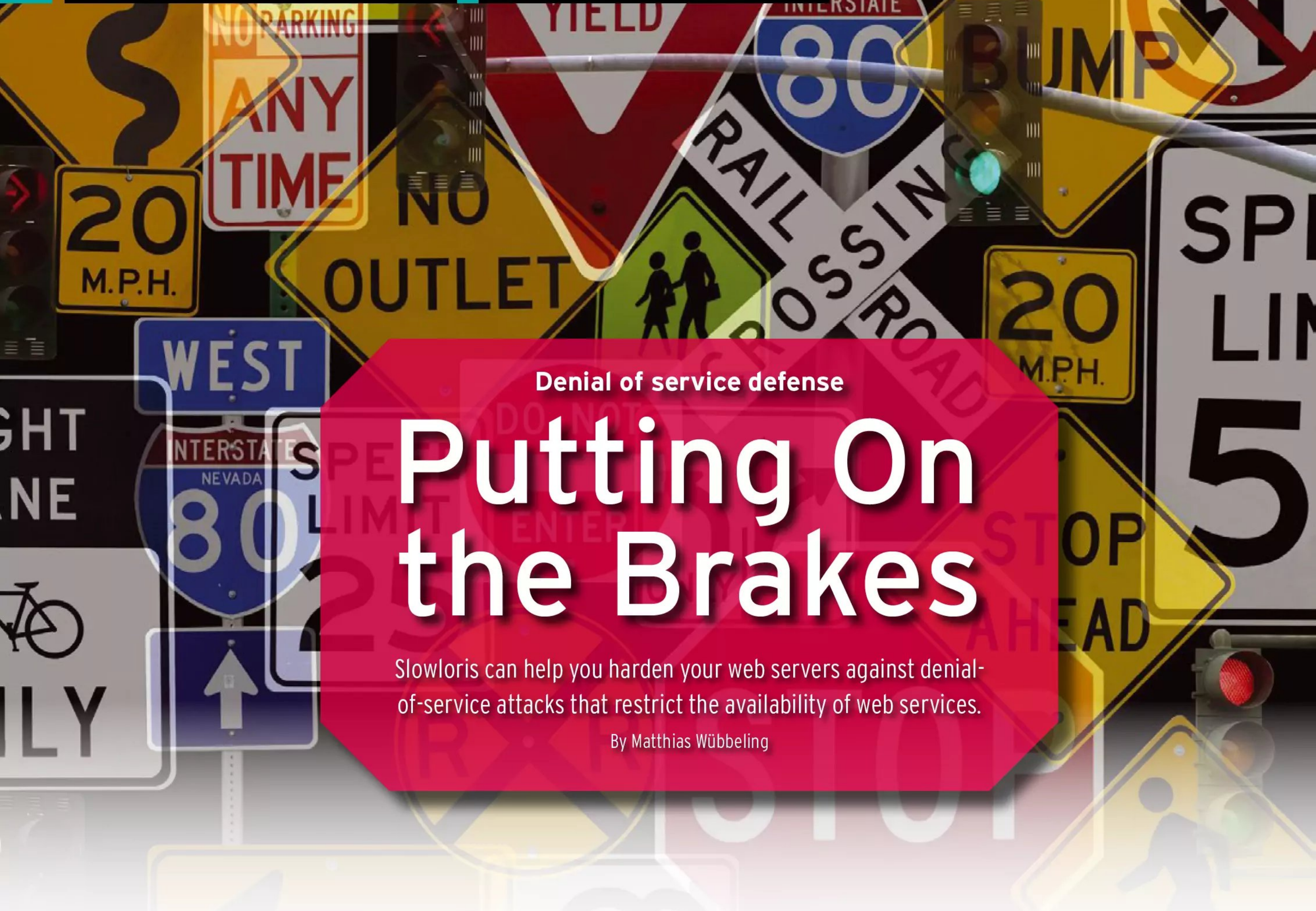
The LibRedirect browser extension redirects YouTube, Instagram, Reddit, TikTok, and other services to alternative, privacy-friendly front ends.

In the News

- System76 Refreshes Meerkat Mini PC
- First Release Candidate for Linux Kernel 6.14 Now Available

In Case You Missed It

- **Simplify VPN Management**
  WireGuard is a less complex VPN solution compared to OpenVPN and IPsec. WireGuard Easy simplifies the process even further by allowing you to operate a VPN and manage clients through a user-friendly web interface.
- **Tools for Enumerating Resources**
  A cyberattack unfolds in stages. The enumeration phase



Denial of service defense

Putting On the Brakes

Slowloris can help you harden your web servers against denial-of-service attacks that restrict the availability of web services.

By Matthias Wübbeling

The Hypertext Transfer Protocol (HTTP) forms the foundation for communication between web browsers and web servers. Slowloris [1] initially follows the usual procedure of an HTTP connection, but then disrupts processing of the request by sending an incomplete HTTP request at a very slow speed, as follows: (1) Slowloris uses TCP to open a client connection to the target server; (2) instead of sending a full HTTP request directly to the server, Slowloris begins a request and then continuously, but very slowly, adds headers without ever completing the request; (3) the server fields all of the header data and waits for the request to complete, keeping the connection alive; and finally, (4) because the server can only process a limited number of simultaneous connections, a large number of slow connections render the server unable to accept connections from other clients.

Immune Out of the Box

Not all modern web servers are susceptible to Slowloris attacks. The default configuration of today's most commonly used web server, NGINX, is basically immune, not just because of the settings, but because of the server architecture. NGINX uses an event-oriented asynchronous architecture that, above all, does not maintain a thread to handle each open connection and only needs a minimal amount of memory for each connection.

The `client_header_timeout` and `client_body_timeout` configuration options let you tell NGINX the intervals at which a client needs to send data when opening a connection. If the client fails to keep pace, NGINX responds with a timeout error with HTTP error code 408. However, this also means that you could make the NGINX server even more vulnerable if you configure it incorrectly.

The best-known web server besides Microsoft's Internet Information Server (IIS; which is rarely used and has a market share south of five percent) is the Apache HTTP daemon. By default, Apache keeps established connections open for a long time and maintains a separate process or thread for each request. Whereas the IIS `headerWaitTimeout` configuration option sets the wait time for header data as a function of how the connection is opened, Apache uses the `TimeOut` directive to look at the time since the last packet was received [2].

Modular Extensions for Apache

Apache version 2.2 and newer has a `mod_reqtimeout` extension that lets you configure the `RequestReadTimeout` option, which you can use to set different request-specific timeouts globally across the server or for individual virtual hosts. The following line tells Apache to wait a maximum of 10 seconds to receive header data; the limit is set at 30 seconds; and the wait is extended by one second for every 500 bytes received:

```
RequestReadTimeout header=10-30, ➥
  MinRate=500
```

If you use the Apache `mod_security` extension, you can also configure the connection-specific parameters.

Lead Image © Russell Shively, 123RF.com

Pentesting a Web Server

If you run a web server or are responsible for web server operations in your organization, you will naturally want to know how it responds in the event of an attack. Before you deploy Slowloris, make sure you have permission to run tests against the target server. In an ideal case, the IT department – or you yourself – will be able to provide a non-production shadow server for initial tests. If this is not possible, you could use Docker to set up one or two web servers on your localhost for your tests. To install the Python implementation of Slowloris on your system, use the command

```
pip3 install slowloris
```

If you cannot install (e.g., because you cannot install the package globally) or if you want to take a look at the code before using it, simply clone the Slowloris repository,

```
git clone ➘
  https://github.com/gkbrk/slowloris.git
```

change to the directory, and run Slowloris when you get there.

Bombs Away

To carry out an initial test with Slowloris, type the command

```
python3 slowloris.py example.com
```

Don't forget to change the domain name from `example.com` to the domain you want to test in your own environment. Of course, you can customize Slowloris with a number of parameters (e.g., relating to the test intensity and type of attack). For example, you can change the number of open connections or the duration between transmitting individual headers and other factors. As you can see, you can work around the defenses deployed on the servers – provided you know the values you set on the servers. For example, if you want to launch an attack with 5,000 connections, you would call Slowloris with the `-s 5000` option. The following call launches an attack against a TLS-protected server on port 443:

```
python3 slowloris.py example.com ➘
  -s 5000 --https -p 8443
```

You might notice a remarkable effect of the Apache configuration at this point: If you have not changed any settings there, you should not be able to open even 1,000 simultaneous connections, because that is the default value for the available connections in Apache. The number of available processes per threads and, in turn, the number of available connections are used up before you reach this limit, which means that legitimate users can no longer reliably connect to the web server until you terminate the above command.

Conclusions

The Slowloris example in this article shows how easy it is to carry out a denial-of-service attack (Table 1) and how you can protect your web server against this attack. If you discover a vulnerability in one of your web servers, no matter how small the effect might seem at first glance, you need to define and roll out possible countermeasures without delay. ■

Info

[1] Slowloris: [https://github.com/gkbrk/slowloris]

[2] Web limits: [https://learn.microsoft.com/en-us/iis/configuration/system.applicationhost/weblimits]

The Author

Dr. Matthias Wübbeling is an IT security enthusiast, scientist, author, consultant, and speaker. As a lecturer at the University of Bonn in Germany and researcher at Fraunhofer FKIE, he works on projects in network security, IT security awareness, and protection against account takeover and identity theft. He is the CEO of the university spin-off Identeco, which keeps a leaked-identity database to protect employee and customer accounts against identity fraud. As a practitioner, he supports the German Informatics Society (GI), administrating computer systems and service back ends. He has published more than 100 articles on IT security and administration.

Table 1: Known Denial-of-Service Attacks

Type of Attack	Description	Resource consumption	DoS/DDoS*
Volume-based attacks	Flood the network with traffic to overload the bandwidth.	Network bandwidth	DDoS
UDP/ICMP flooding	Send a large number of UDP/ICMP packets to overload the target system.	Network bandwidth, CPU	DDoS
SYN flooding	Send a large number of SYN requests without completing the handshake to block connections.	Network resources, memory	DDoS
Ping of death	Send oversized or fragmented ping packets that cause the target system to crash.	Network bandwidth, memory	DoS
Slowloris	Keep HTTP connections open because of slow transmission of header data to reach the maximum connection limit.	Server connections, memory	DoS
HTTP slow POST/RUDY	Send very slow POST data to bind server resources.	Server connections, CPU, RAM	DoS
Reflection	Send requests to third-party servers with a spoofed sender address so that the responses flood the victim.	Network bandwidth	DDoS
Amplification	Exploit open servers (DNS, NTP, SSDP) to flood the victim's bandwidth with large volumes of data.	Network bandwidth	DDoS

*DoS, denial of service; DDoS, distributed denial of service; UDP, User Datagram Protocol; ICMP, Internet Control Message Protocol; RUDY, R U dead yet; DNS, domain name system; NTP, Network Time Protocol; SSDP, Simple Service Discovery Protocol.

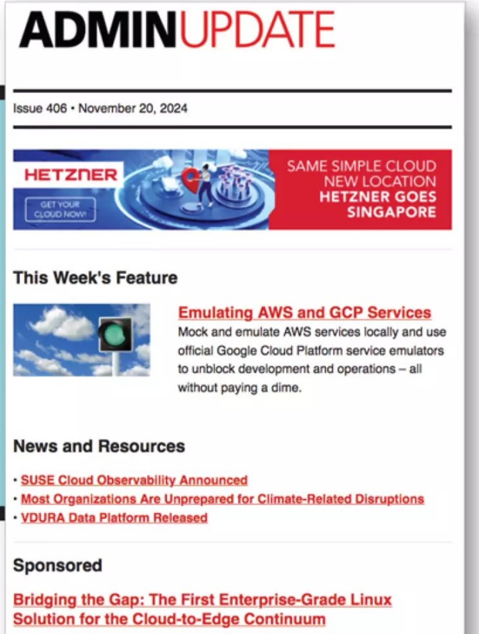
ADMIN Network & Security
Zero Trust
grommunio
Storage
ISSUE 83
STORAGE
Developments, management, and new tech
Deduplication on Windows Server 2022
Non-Relational Databases
Monster VMs
Gitea
ISSUE 84
Non-Relational Databases
NoSQL, vector, and key-value stores
Purdue Model
Kubeflow
Monitoring
ISSUE 85
Monitoring
Networks, applications, and long-term storage
Bloonix
Web-based network monitoring
Kubeflow
Kubernetes setup for AI development
Azure Dev Box Cloud Service
Prometheus Long-Term Storage
Attack Surface Management
Tools and strategies for threat management
Purdue Model
Map the challenges of industrial systems
Podman Quadlets
Run rootless containers as systemd services
Julia
A backbone for interactive websites
DVD INSIDE
WWW.ADMIN-MAGAZINE.COM

Want to get ADMIN in your inbox?

Subscribe free to ADMIN Update

and get news and technical articles you won't see in the magazine.

bit.ly/ADMIN-Update



ADMINUPDATE

Issue 406 • November 20, 2024

HETZNER
GET YOUR CLOUD NOW!
SAME SIMPLE CLOUD
NEW LOCATION
HETZNER GOES SINGAPORE

This Week's Feature

Emulating AWS and GCP Services

Mock and emulate AWS services locally and use official Google Cloud Platform service emulators to unblock development and operations – all without paying a dime.

News and Resources

- SUSE Cloud Observability Announced
- Most Organizations Are Unprepared for Climate-Related Disruptions
- VDURA Data Platform Released

Sponsored

Bridging the Gap: The First Enterprise-Grade Linux Solution for the Cloud-to-Edge Continuum

Get to know

ADMIN

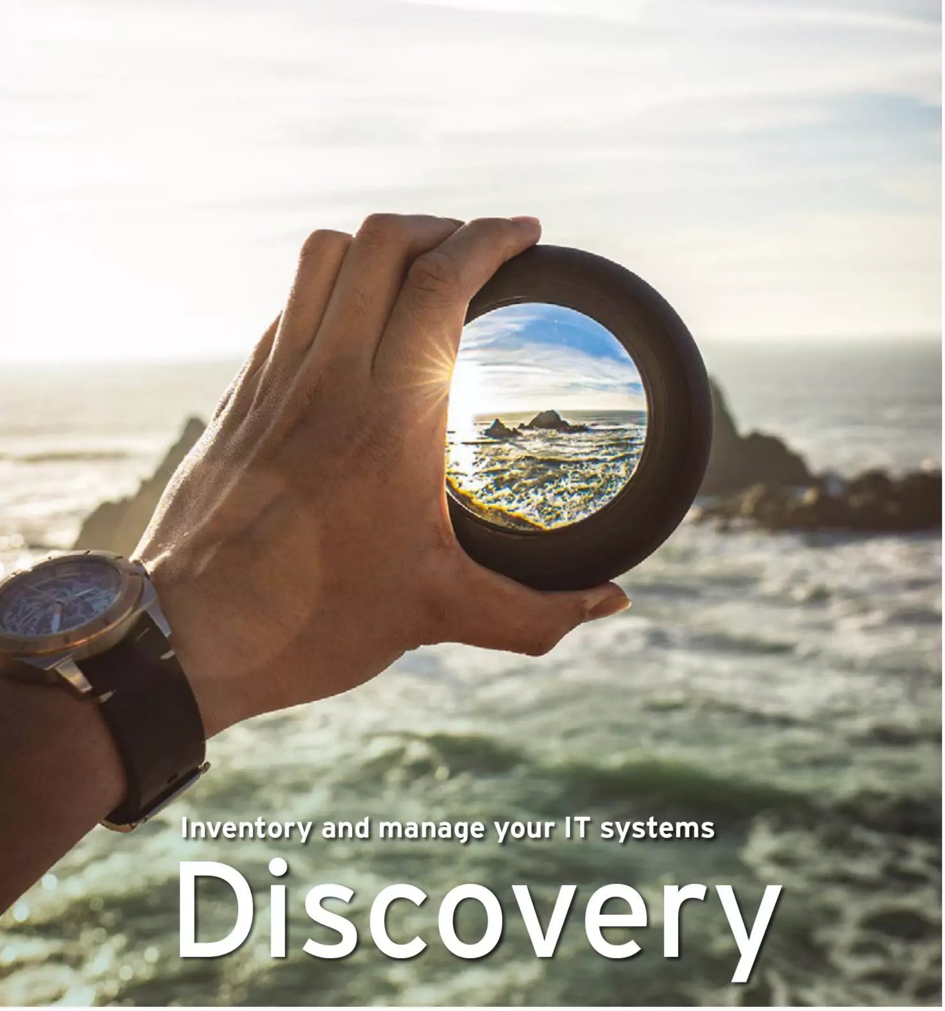
ADMIN Network & Security **magazine is your source for technical solutions to real-world problems.**

ADMIN is packed with detailed discussions aimed at the professional reader on contemporary topics including security, cloud computing, DevOps, HPC, containers, networking, and more.

Subscribe to *ADMIN*
and get 6 issues
every year

ADMIN
Network & Security

ADMIN magazine
@adminmag
@adminmagazine@hachyderm.io



Inventory and manage your IT systems

Discovery

Open-AudIT inventories, manages, and monitors IT assets, whether hardware, software, network device, or virtual machine. We explain how to install the software and work through the initial configuration.

By Thorsten Scherf

Open-AudIT helps you to discover all of your IT resources. The software automatically identifies the devices and applications on your network, giving you up-to-date information at any time, which, in turn, is a massive help for compliance policies and security guidelines. Open-AudIT also has an option for continuous monitoring of your network, not only with a view to detect unauthorized devices or applications, but also to identify and shut down potential vulnerabilities. The intuitive user interface does not require in-depth technical knowledge, and the installation and setup process is straightforward.

Comprehensive reporting functions then help you make day-to-day decisions relating to your IT assets. You can create user-defined reports to analyze specific information and adapt your IT strategy according to the results. The software integrates with the existing system landscape and can be adapted to your organization's specific needs.

Long History

The first, still very simple, and scripting-based version of Open-AudIT was published by Mark Unwin back in 1999. After a complete overhaul of the software in PHP, Open-AudIT was released under the GNU General Public License in 2006 and has seen many additional features added, including a web interface. Thanks to the project's great popularity, Australia's Opmantek acquired it in 2012 and released several major releases in the following years until the company was finally acquired by cybersecurity and network management specialist FirstWave in 2021. This acquisition put FirstWave in a position to add Open-AudIT's IT asset management and network audit capabilities to its portfolio and offer customers a more comprehensive suite of IT management tools. The most recent version 5.6.2 was released in January 2025 by Mark Unwin, who is still actively involved in the project.

The software is available in the form of the Community, Professional, Enterprise, and Cloud editions. The codebase of these editions is fundamentally identical, but the Community version lacks some plugins and functions that are included in the other releases. Of course, commercial support is not available for this version; instead, a very active community is happy to help with problems.

Different Installer Types

If you want to try out the Community version of the software, you will probably end up on the Open-AudIT project's website **[1]**. The *Downloads* link takes you to a website where you can select an installation file for either Linux or Windows. However, it is somewhat confusing that a number of links still point to the old Opmantek website, even though the company was acquired by FirstWave more than three years ago. It seems that the Opmantek website is primarily intended for members of the community, whereas the FirstWave website **[2]** clearly has enterprise customers in mind.

FirstWave also offers a software appliance for Open-AudIT, and an image file for operations on a virtual machine (VM). The Network Management Information System (NMIS) it contains includes both Open-AudIT and other products, such as opCharts, opConfig, opEvents, and opReports, to name a few. The Debian-based appliance has options for comprehensive network management in the form of a preconfigured VM. For more details, visit the product page **[3]**.

Photo by Edryc James P. Binoya on Unsplash

The work in this article is based on the current community version of Open-AudIT at the time of writing (v5.2.2, released in May 2024). Apart from the software appliance, the installation is no different from that of the other the software editions and the basic configuration is identical.

Installation with Minor Setbacks

For this article, I chose a recent Red Hat Enterprise Linux (RHEL) v9.4, but you can also run the Linux installer on other recent Linux distributions such as Debian or Ubuntu. If some of the required packages are missing, the installation script will automatically download and install them from your distribution's software repository. Integrating external repositories is also part of the installation process. On a RHEL system, for example, the well-known Extra Packages for Enterprise Linux (EPEL) repository is automatically integrated by the installer, and packages are installed from it. Of course, you are first prompted to confirm that you want this to happen (**Figure 1**).

You can launch the installer with the help of sudo:

```
# sudo ./OAE-Linux-x86_64-release_5.2.2.run
```

After the install, you will find the files in the `/usr/local/open-audit` directory. The `public` sub-folder is used as the document root of the Apache web server. Note that I was unable to run the software on RHEL in SELinux enforcing mode in the lab. In fact, I had to switch to permissive mode to exit the installer without error. There is certainly room for improvement here. Alternatively, you have the option of creating a separate SELinux policy module for the software to continue operating the system in secure SELinux enforcing mode.

The setup installs and configures a MariaDB database and an Apache web server. Apache will be responsible later for storing all the details of your IT assets. Both the database and the username for the installer is *openaudit*. The user password is displayed on the screen during the installation and is also included in the documentation.

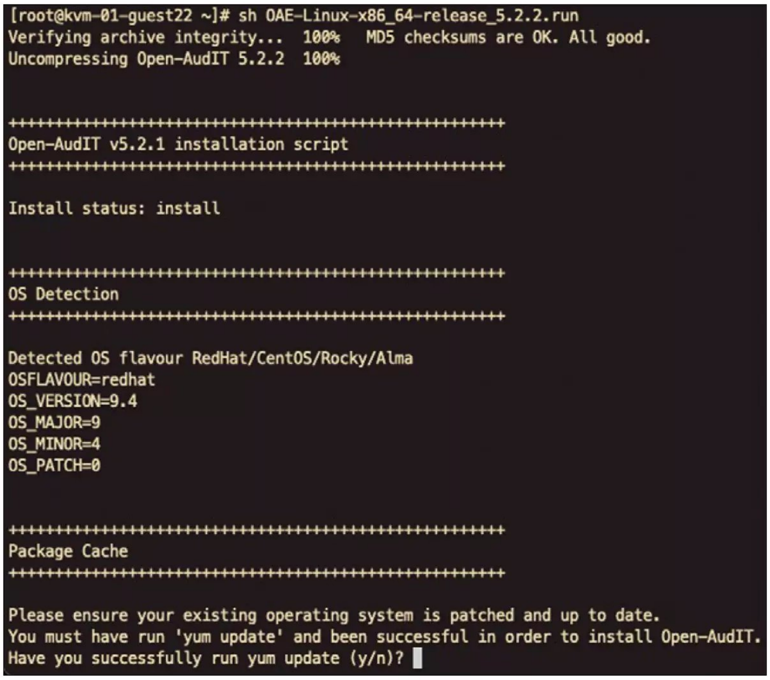
Watch out for a few quirks if your are installing on Windows. Open-AudIT runs smoothly as a 64-bit app on a current Windows Server, but not on Windows 10 or 11. As on Linux, you need the Nmap network scanner, which is used during the network device discovery process. Although this package is installed automatically on Linux, Windows users first need to download the Nmap software from the website **[4]** and then install it manually. A Visual C runtime **[5]** also needs to be installed up front. With the underpinnings in place, you can start to install Open-AudIT working in the Administrator user context. The installer is available for download as an EXE file **[1]**.

Web Interface

After completing installation, it is time to take a look at the software's web interface. To do this, simply type the name of the system on which you ran the installer in your browser followed by *open-audit* (e.g., *http://monitoring.example.com/open-audit/*) and log in as *admin* user. The password for the initial login is displayed on the login page. If you want to test the functions of the Enterprise version now, you can purchase a time-bombed one-year license in the *Licenses* drop-down. Note that, although you will have access to more functions than in the Community version in this case, the number of manageable devices is limited to 20. In this respect, purchasing this license is really only useful for test purposes. If you require the feature set of the Enterprise Edition, I recommend purchasing a commercial license, which you can import from the *Licenses* drop-down, as well. A detailed overview of the available functions for each edition of Open-AudIT is shown online **[6]**.

Initial Configuration

Regardless of whether you are using the Community or Enterprise edition, you will want to start by configuring some basic settings, including the email system, which is important for delivery of messages, such as reports from the Open-AudIT discovery processes. You can configure the settings in the *Admin* | *Configuration* menu.



```
[root@kvm-01-guest22 ~]# sh OAE-Linux-x86_64-release_5.2.2.run
Verifying archive integrity...  100%   MD5 checksums are OK. All good.
Uncompressing Open-AudIT 5.2.2  100%


++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Open-AudIT v5.2.1 installation script
++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Install status: install


++++++++++++++++++++++++++++++++++++++++++++++++++++++++
OS Detection
++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Detected OS flavour RedHat/CentOS/Rocky/Alma
OSFLAVOUR=redhat
OS_VERSION=9.4
OS_MAJOR=9
OS_MINOR=4
OS_PATCH=0


++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Package Cache
++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Please ensure your existing operating system is patched and up to date.
You must have run 'yum update' and been successful in order to install Open-AudIT.
Have you successfully run yum update (y/n)? 
```

Figure 1: The Open-AudIT script-based installer guides you through the software setup process.

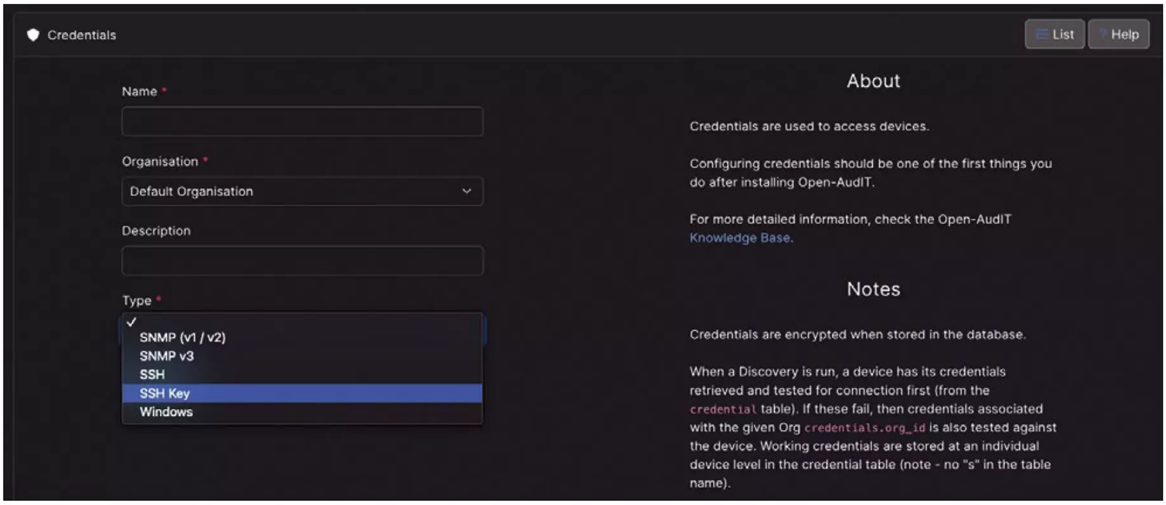
The Enterprise Edition has more than 160 different configuration options. For an overview of the default settings go to *Admin* | *Configuration* | *List Configuration*.

The next step is to store the credentials for access to your IT assets, which are essential for the discovery process that retrieves information about your devices and for displaying the results in Open-AudIT. The *Discover* | *Credentials* | *Create Credentials* menu item gives you the option of entering access credentials for Simple Network Management Protocol (SNMP), secure shell (SSH), or Windows systems. For SSH-based access, you can of course use SSH keys in addition to passwords (Figure 2). Open-AudIT supports SNMP versions 1, 2, and 3 and uses the Windows Management Instrumentation (WMI) specification to access Windows systems to discover information on the hardware and software in use.

Open-AudIT stores the credentials in encrypted form in a MariaDB table named *Credentials*. By default, however, the details will be saved in the clear in a JSON or CSV file when the access credentials are exported. This arrangement is not desirable; therefore, you are advised to change the `decrypt_credentials` configuration parameter from `yes` to `no`. Now, when you export the data, it will be in the encrypted form, just as in the database itself.

Network and Active Directory Scans

Credentials alone are useless if you cannot also use them as part of a discovery or audit process. At the end of the day, Open-AudIT is essentially about identifying devices on a network and serving up information


Credentials
List
Help
Name *
Organisation *
Default Organisation
Description
Type *
✓
SNMP (v1 / v2)
SNMP v3
SSH
SSH Key
Windows

About

Credentials are used to access devices.

Configuring credentials should be one of the first things you do after installing Open-AudIT.

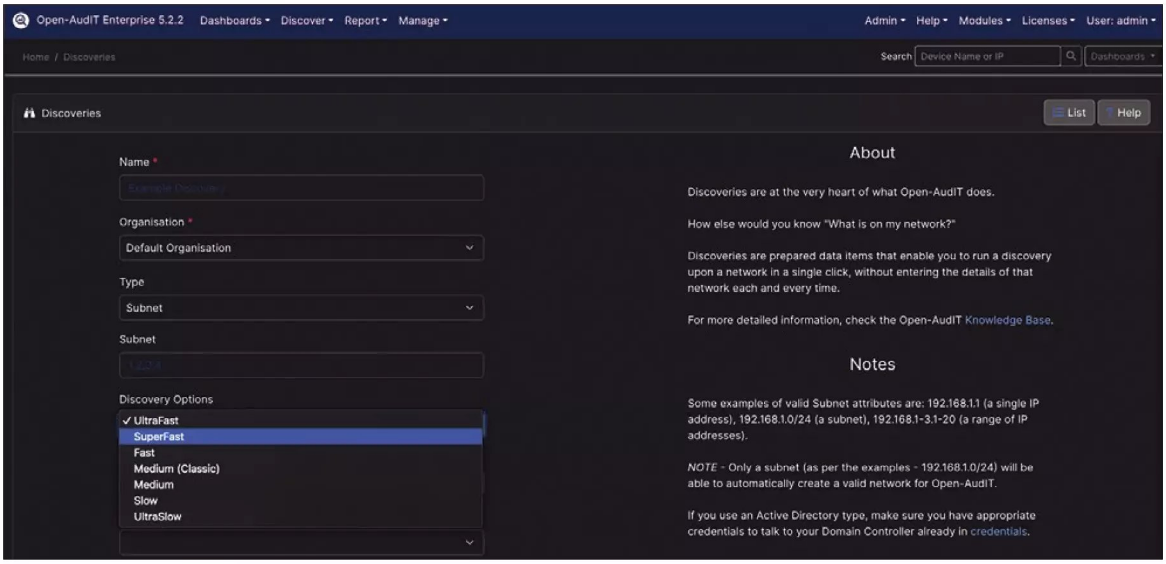
For more detailed information, check the Open-AudIT Knowledge Base.

Notes

Credentials are encrypted when stored in the database.

When a Discovery is run, a device has its credentials retrieved and tested for connection first (from the `credential` table). If these fail, then credentials associated with the given Org `credentials.org_id` is also tested against the device. Working credentials are stored at an individual device level in the credential table (note - no "s" in the table name).

Figure 2: For more extensive system audits, Open-AudIT requires the access credentials for your systems, which are then encrypted and stored in a database.


Open-AudIT Enterprise 5.2.2
Dashboards
Discover
Report
Manage
Admin
Help
Modules
Licenses
User: admin
Home / Discoveries
Search
Device Name or IP
Dashboards
Discoveries
List
Help
Name *
Organisation *
Default Organisation
Type
Subnet
Subnet
Discovery Options
✓ UltraFast
SuperFast
Fast
Medium (Classic)
Medium
Slow
UltraSlow

About

Discoveries are at the very heart of what Open-AudIT does.

How else would you know "What is on my network?"

Discoveries are prepared data items that enable you to run a discovery upon a network in a single click, without entering the details of that network each and every time.

For more detailed information, check the Open-AudIT Knowledge Base.

Notes

Some examples of valid Subnet attributes are: 192.168.1.1 (a single IP address), 192.168.1.0/24 (a subnet), 192.168.1-3.1-20 (a range of IP addresses).

NOTE - Only a subnet (as per the examples - 192.168.1.0/24) will be able to automatically create a valid network for Open-AudIT.

If you use an Active Directory type, make sure you have appropriate credentials to talk to your Domain Controller already in credentials.

Figure 3: Open-AudIT uses discovery processes to scan the network.

on them. With *Discover* | *Discoveries* | *Create Discoveries*, you can define different types of network scans. A scan covers a single IP address, an IP range, or an IP subnet.

The type of scan lets you specify the depth of detail with which Open-AudIT investigates a system, which naturally has an effect on how much information is displayed for a particular device (Figure 3). The more detailed a scan, the longer it takes to complete. Creating different scan types for the same network makes sense. For example, one scan could provide a basic overview of the existing systems, and another could provide detailed information on these systems.

If you want to scan the subnets of a Windows domain, Open-AudIT offers the *Active Directory* scanning type for this purpose. Select this parameter in the *Create Discovery* menu item and then enter the domain name and a domain controller for this domain. As soon as you start the discovery process, Open-AudIT automatically discovers all the IP subnets that belong to the specified domain and then performs a scan of the systems on these networks. It is important that the WMI ports of your Windows systems be accessible; if not, Open-AudIT cannot collect any detailed information about the systems.

Starting Discoveries

Once you have defined all the discovery processes for your networks, you can access a complete list of options from the *Discover* | *Discoveries* | *List Discoveries* and then launch discoveries directly from this page with the *Execute* button. The *Logs* option lets you follow the scan process. The information collected on the devices discovered on a network is then accessible from *Details* (Figure 4).

If the results meet your expectations, you can start the individual discovery processes automatically at specific times by going to *Admin* | *Tasks* | *Create Task* and defining which discovery process you want to run automatically and at what time. Also useful is the ability to create tasks to send different reports, so that, besides viewing the recent reports on the dashboard in the Open-AudIT web interface, you can also mail them, if required. When you define a new task, simply select the *Report* task type instead of *Discovery* and define the type of report that you want to mail automatically.

I used the standard administrator account (*admin*) for all of the activities tried for this article. Of course, Open-AudIT offers a role-based access model, so you can create additional users and assign specific authorizations to them. To create a new user account, go to *Manage* | *Users* | *Create Users* and assign it the *admin* role, so the account can handle administrative tasks.

Conclusions

Open-AudIT is a powerful management and information tool for complex networks. In this article, I only scratched the surface, showing you how to install the software and illustrating the initial configuration steps, including creating access credentials and managing the discovery processes, which you can use to scan your networks to glean information on the systems that live there.

If you want to delve deeper into the subject, I recommend a visit to the Open-AudIT support forum [7]. Besides the documentation, you will also find practical articles there to help you implement specific actions. You can use the project's GitHub repository [8] to follow ongoing development of the software or send pull requests to the community. ■

Info

[1] Open-AudIT: [https://www.open-audit.org]

[2] FirstWave: [https://firstwave.com/products/network-discovery-and-inventory-software/]

[3] NMIS: [https://firstwave.com/products/network-management-information-system/]

[4] Nmap network scanner: [https://nmap.org]

[5] Visual C runtime: [https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170]

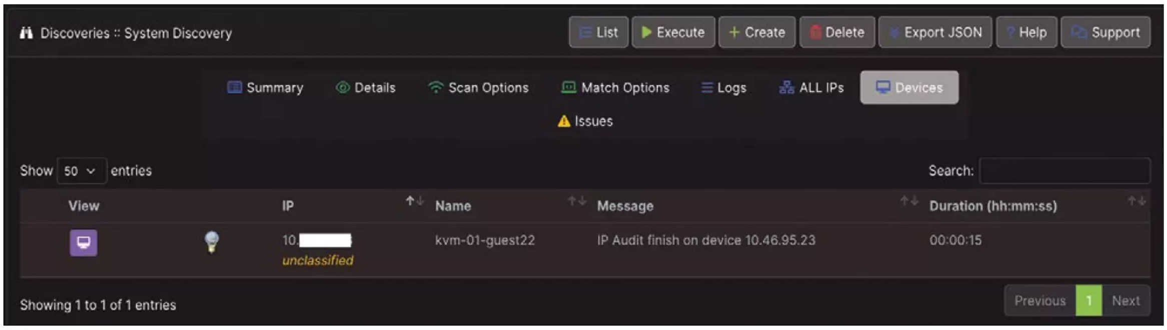
[6] Open-AudIT editions: [https://www.open-audit.org/about.php]

[7] Open-AudIT community support: [https://docs.community.firstwave.com/wiki/spaces/OA/overview]

[8] Open-AudIT GitHub repository: [https://github.com/Opmantek/open-audit]

The Author

Thorsten Scherf is the global Product Lead for Identity Management and Platform Security in Red Hat's Product Operations group. He is a regular speaker at various international conferences and writes a lot about open source software.



View		IP	Name	Message	Duration (hh:mm:ss)
🖥	💡	10.[redacted] *unclassified*	kvm-01-guest22	IP Audit finish on device 10.46.95.23	00:00:15

Showing 1 to 1 of 1 entries

Figure 4: **Open-AudIT delivers information for each discovery process and assigns a version number to each set.**

Open source ticket system

A Helping Hand

OTOBO is a user-friendly and completely free ticket system designed to handle complex service requirements that offers an alternative, easy-to-use support tool with Active Directory integration and single sign-on. We show you how to set up and configure OTOBO and how to connect it to third-party systems such as Entra ID. By Tobias Bück

From the outset, OTOBO's developers designed for efficiency, adaptability, and visual appeal. With features such as Docker support, dynamic forms, and full-text search, the software enables fast information extraction and can be customized to your organization's needs. The free software makes full use of the possibilities of modern IT service management systems and offers a platform for an active community. The help desk system aims to be a cost-efficient technological alternative to proprietary products, to which its sophisticated features contribute.

Architecture and Functions

OTOBO [1] offers a range of advanced features for efficient and effective ticket processing. One of its key features is its ability to connect to Active Directory (AD) and other services that use the Lightweight Directory Access Protocol (LDAP) to support centralized management of user accounts and group memberships. This integration simplifies user administration, authentication, and authorization. Best practices for integration include multiple backup LDAP servers for redundancy, minimized querying through targeted search and filtering, and regular reviews of access rights with a view to data security and compliance.

The platform also comes with secure email communication that supports OAuth 2.0 for POP3 and IMAP email protocols to enable modern and secure authentication for the retrieval of email from services such as Microsoft 365. The required OTOBO daemon automates the retrieval process by collecting email messages at predefined intervals and automatically running processing tasks. Also important is the configuration of postmaster filters for the automatic assignment of email to the correct queues.

The integration of single sign-on (SSO) means that users only need to authenticate once with their credentials to gain access to multiple systems, including OTOBO, without having to log in repeatedly. This feature enhances usability and security. Before moving on to the details of installing and operating the ticket system, I'll look at the new features in the major release 11 published in May 2024.

New in OTOBO 11

Release 11 of OTOBO offers a number of new dynamic fields designed to offer greater flexibility and performance. One particularly interesting candidate is the new Script field, which can use logical operators to check conditions or perform calculations. Two examples of how this field is useful are ensureing that orders comply with budget specifications and determining the warranty status of products. The *DynamicFieldSet* parameter lets you group several dynamic fields in a set, which considerably accelerates the process

Photo by Toa Heftiba on Unsplash

of configuring complex forms. The *MultiValue* function allows dynamic fields or sets to be added or removed at runtime, which is particularly useful for entering multiple items with quantities and prices.

The ticket screens in version 11 can be customized even more and more clearly than in the past. Dynamic fields can be provided in different orders on different screens and with individual labels. The option of arranging dynamic fields in a grid with an arbitrary number of columns further improves flexibility, and dynamic fields and sets can be added or removed at runtime by the *MultiValue* function, which in turn enables flexible customization. Read-only fields with default values on selected screens are also helpful in processes that offer information from upstream steps as a reference.

The operation of multilingual systems is also more convenient. System administrators can now maintain translations directly in the admin interface instead of in hidden system files. Convenient filter and display options let you run targeted searches for untranslated elements in the system. Terms can either be translated directly in the interface or exported to an Excel list and reimported in translated form, making the process far easier.

OTOBO 11 also offers the option of subsequently editing and deleting internal articles. The new CKEditor 5 is designed to ensure user-friendly text editing. Ticket wait times can be individually configured and adjusted by the new QuickTime buttons. The *Customer Info Tile* allows you to create rich-text tiles for the customer area, which are managed in the admin interface. Support can also add tickers for particularly urgent messages to these tiles if needed.

Installing OTOBO

With support for Docker and Docker Compose, OTOBO offers great flexibility and an easy installation process across a wide range of operating systems. Docker is required to operate OTOBO on Windows servers; a native installation is only supported on Linux machines. Containers let you run the software in isolated environments to minimize compatibility and dependency problems. Docker Compose simplifies the process by enabling the deployment of multicontainer Docker applications with a single configuration file, which automatically sets up and configures all required OTOBO services.

OTOBO hardware and software requirements vary depending on the deployment scenario, from minimal installations to systems optimized for high-volume ticket processing. When you are planning your OTOBO installation, take a look at the guidelines **[2]**. The free disk space requirements can vary depending on the number of tickets and the size of the attachments. Performing regular backups can prevent potential data loss. If incoming email messages often have large attachments, your storage requirements can grow quickly.

Before you start the installation, make sure Docker and Docker Compose are installed. This process can vary depending on your choice of operating system; on Ubuntu it would be

```
sudo apt update && sudo apt install ↩
  docker.io docker-compose
sudo systemctl enable --now docker
```

Next, clone the official OTOBO Docker repository, which contains all the Docker Compose files and configurations you need:

```
cd /opt
git clone https://github.com/↩
    RotherOSS/otobo-docker.git ↩
  --branch rel-10_1 --single-branch
cd otobo-docker
```

The `--branch rel-10_1` option indicates the version; versions 11 (`--branch rel-11_0`), 10.1 (`--branch rel-10_1`), and 10.0 (`--branch rel-10_0`) were available at publication. To install specific patch versions, enter `--branch rel-10_1_<X>`.

The primary management interface is the Docker Compose ENV configuration file, which you first need to create and then customize. To simplify the task, you can refer to several sample files as your starting point. Which to choose depends on your use case. In most cases, the decision will be between `docker_compose_env_http` and `docker_compose_env_https`, depending on whether you need TLS support. The other files are for more specific use cases such as HTTP, HTTPS with NGINX running as a reverse proxy web server, or SSO. Kerberos support is available but still tagged as experimental.

By way of an example, to create a configuration file for HTTPS, use

```
cd /opt/otobo-docker
cp -p .docker_compose_env_https.env
```

Then, go to the directory in which your `docker-compose.yml` and ENV files reside, change the password setting, and start the OTOBO containers with Docker Compose:

```
OTOBO_DB_ROOT_PASSWORD=<password>
nano .env
docker-compose up -d
```

The command downloads the images you need and starts the containers; use `docker ps` to check that all containers are in operation.

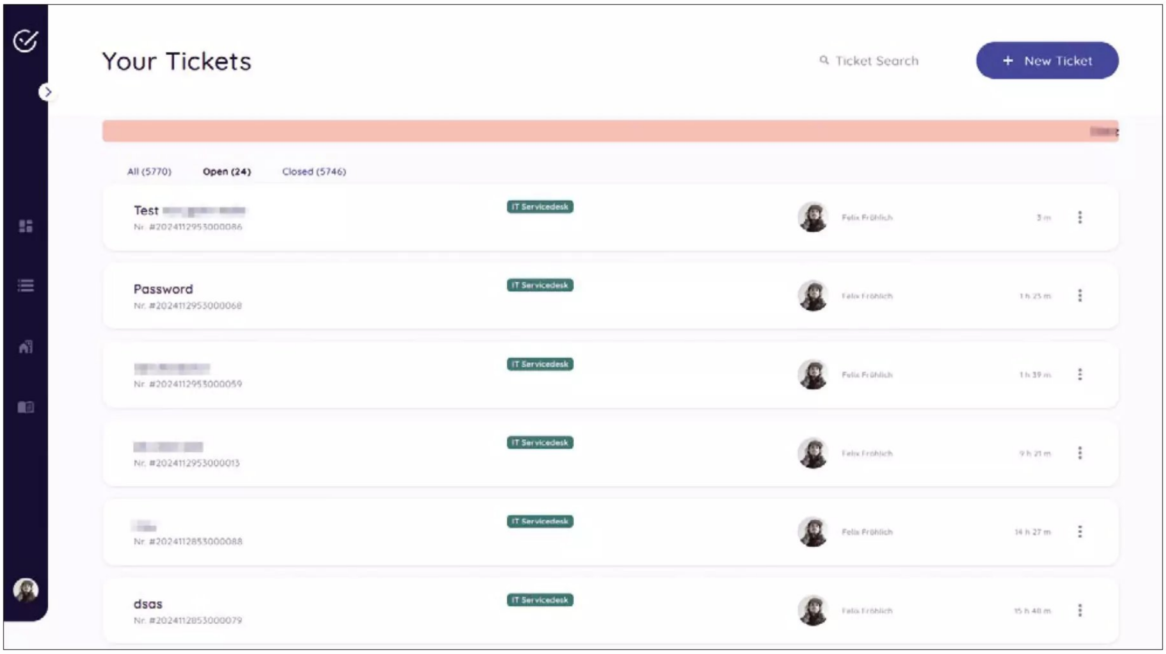
With the containers running, you can now access the OTOBO installer from a web browser at *http:///otobo/installer.pl* and follow the instructions in the web installer. Among other things, you can configure the database and define the admin account. Once you have completed the configuration in the web installer, OTOBO is ready for use. You can now log in to the admin area and configure your ticket system to suit your organization's needs.

Exploring OTOBO

The core elements of OTOBO (agents, queues, groups, customers, customer users, tickets, types, priorities) ensure a smooth and organized customer support process and enable precise control of the service processes. The agents are the system users who process the tickets and communicate with customers. OTOBO supports agent management with various back-end sources, offering extensive integration options for managing the authorization system.

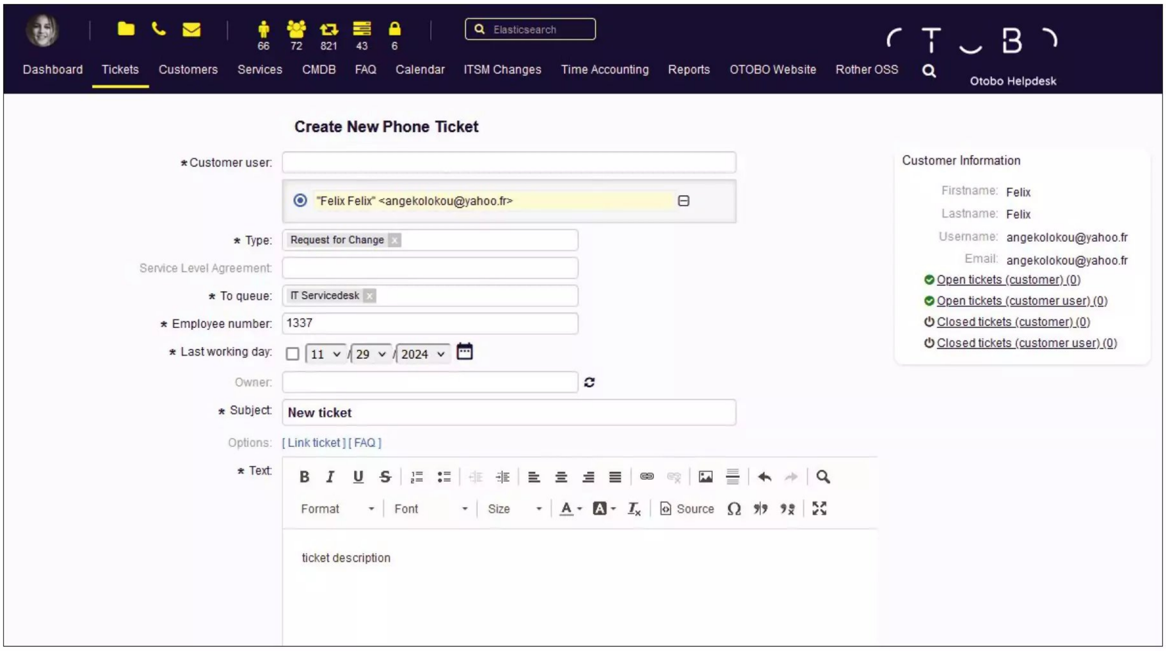
Queues organize the tickets and assign them to teams by skill levels, fields of responsibility, or departmental affiliations (Figure 1). Groups manage access and authorizations. When you assign agents to groups, you are coordinating access to tickets, functions, and areas within the system to create a secure and organized work environment.

You can also manage customers and their users (customer users). This structure allows the processing of requests centrally and ensures that each finds the correct handler. The ability to use different data sources (e.g., databases and LDAP directories) for customer management massively boosts flexibility.


Your Tickets
Ticket Search
+ New Ticket
All (5770)
Open (24)
Closed (5746)

Test [redacted] Nr. #2024112953000086	IT Servicedesk	Felix Fröhlich	3 m	⋮
Password Nr. #2024112953000068	IT Servicedesk	Felix Fröhlich	1 h 25 m	⋮
[redacted] Nr. #2024112953000059	IT Servicedesk	Felix Fröhlich	1 h 39 m	⋮
[redacted] Nr. #2024112953000013	IT Servicedesk	Felix Fröhlich	9 h 21 m	⋮
[redacted] Nr. #2024112853000088	IT Servicedesk	Felix Fröhlich	14 h 27 m	⋮
dsas Nr. #2024112853000079	IT Servicedesk	Felix Fröhlich	15 h 48 m	⋮

Figure 1: The OTOBO customer portal provides an overview of the current tickets.


66
72
821
43
6
Elasticsearch
Dashboard
Tickets
Customers
Services
CMDB
FAQ
Calendar
ITSM Changes
Time Accounting
Reports
OTOBO Website
Rother OSS
Otobo Helpdesk
Format
Font
Size
Source

Create New Phone Ticket

* Customer user:
● "Felix Felix" <angekolokou@yahoo.fr>
* Type: Request for Change
Service Level Agreement:
* To queue: IT Servicedesk
* Employee number: 1337
* Last working day: ☐ 11 / 29 / 2024
Owner:
* Subject: New ticket
Options: [Link ticket] [FAQ]
* Text: ticket description

Customer Information
Firstname: Felix
Lastname: Felix
Username: angekolokou@yahoo.fr
Email: angekolokou@yahoo.fr
Open tickets (customer) (0)
Open tickets (customer user) (0)
Closed tickets (customer) (0)
Closed tickets (customer user) (0)

Figure 2: In this screen, you create new tickets on the basis of a user's phone number.

Tickets are the basis of communication in OTOBO. They record every request, problem, or case created by customers or created internally (Figure 2). Dynamic fields allow for detailed information on tickets, which in turn boosts data analysis and reporting capabilities. You can define ticket types (e.g., *unclassified*, *incident*, or *problem*) to categorize requests. Moreover, the system uses a traffic light system to help you define priorities, ensuring that urgent requests receive immediate attention. In the default installation, tickets can have one of five priorities, but it is often sufficient to use just three.

Assigning agents to groups is essential for the efficient processing of tickets and requests. These assignments can include finely tuned permissions ranging from read-only access to full access and even specific actions such as entering notes or editing ticket priorities.

Microsoft 365 Link

The integration of email from Microsoft 365 into OTOBO by OAuth 2.0 offers a secure and state-of-the-art method for email retrieval. Getting this to work involves a few key steps, from setting up in Entra ID to configuring in OTOBO. I'll begin on the Entra ID side first:

1. Create an enterprise application. To begin, you need to create an enterprise application on the portal that represents your OTOBO instance in the Microsoft cloud and serves as the basis for authentication.
2. Create a redirect URL: For OAuth 2.0 authentication, you need to define a redirect URL that routes back to the OTOBO instance. This URL comprises the address of your OTOBO installation followed by */otobo/index.pl?Action=AdminMailAccount*.
3. Generate a secret client key: You need to create this key in Azure to enable later authentication in OTOBO.
4. Add API authorizations: The app requires specific authorizations to access the email data. Authorizations for `IMAP.Access-AsUser.All` and `POP.AccessAsUser.All` are important.

After the setup in Entra ID, the next step is to complete the configuration in the OTOBO ticket system:

1. Install the *MailAccount-OAuth2* package: In OTOBO, use the package manager to install this package to extend OTOBO to include the functionality it needs to use OAuth 2.0 for retrieving email.
2. Customize the OTOBO system configuration: In the system configuration, you need to enable the `OAuth2::MailAccount::Profiles###Custom1` profile and configure the application ID, client key, and tenant ID. You also need to enter the specific

What?!
I can get my issues SOONER?
Available anywhere, anytime!

Sign up for a digital subscription to improve our admin skills wth practical articles on network security, cloud computing, DevOps, HPC, storage and more!

Subscribe to the PDF edition:
shop.linuxnewmedia.com/shop

TenantID instead of *common* in the AuthURL and TokenURL sections to ensure successful authentication.

3. Select the PostMaster mail account profile: Create an email account (or customize an existing one) by selecting the previously configured *OAuth2* profile so the PostMaster process can use OAuth 2.0 to retrieve email from the Microsoft 365 account.

Make sure all required API authorizations are granted in Azure and your firewall is configured to allow communication with the Microsoft 365 servers. Without these steps, authentication could fail or email retrieval could be blocked.

Connect Active Directory

Integrating users from Active Directory offers an efficient approach to centralizing and simplifying user management in large organizations. After connecting an LDAP or Active Directory server, you can seamlessly transfer user data to the OTOBO system, eliminating the need for manual maintenance.

To begin, you need to prepare the LDAP back end in OTOBO for integration with Active Directory. The LDAP back end is configured in the `Kernel/Config.pm` configuration file, in which you store the LDAP server details. Please note that OTOBO's files are located in volumes because of the Docker installation, which explains why you will find the files in `/var/lib/docker/volumes/otobo_opt_otobo/_data`.

To customize the configuration, begin by copying the `$Self->{CustomerUser}` section from the `Kernel/Config/Defaults.pm` file to the `Kernel/Config.pm` file and remove the hashes (`#`) to enable the required LDAP settings, including the server address (`Host`), base domain name (`BaseDN`), and user identification (`UID`), which is typically the `sAMAccountName` for AD integration.

Next you need to adjust the attribute mappings between LDAP attributes and OTOBO-specific fields to ensure flexible handling of customer data, such as the assignment of the full name, email address, and other relevant customer data.

Additionally, OTOBO supports parallel use of different back ends to help you manage user data from multiple sources. If you number the back ends (e.g., `CustomerUser1`, `CustomerUser2`, etc.) in the `Kernel/Config.pm` configuration file, the ticket system can integrate up to 10 different back ends.

Active Directory integration also enables granular control of user authorizations within OTOBO. By assigning AD groups to OTOBO roles, you can map extensive access authorizations on the basis of existing company guidelines. Ultimately, this interaction ensures seamless synchronization of user data between the systems, reduces the administrative overhead involved in user management, and improves security through centralized authentication mechanisms.

Enabling SSO with AD

The implementation of single sign-on by Kerberos in OTOBO enables seamless and secure authentication, especially in AD environments. This process massively simplifies the login experience for users, eliminating the need to re-enter credentials by automatically logging them in with their Microsoft credentials.

For SSO, you first need to create an AD user account that will be used for SSO authentication. Assigning usernames that follow a specific pattern, such as *HTTP/<domain>*, is important. Make sure you use capital letters for the *HTTP/* part (as expected by Kerberos) and ensure the domain name is an A record and not a CNAME record. Now use the `ktpass` tool on a domain controller to create a `.keytab` file for the user you just created; this file is needed later in the OTOBO configuration to enable Kerberos authentication.

Next, you need to prepare NGINX and the OTOBO containers by creating a new directory and a volume for the NGINX configuration on your OTOBO server. Move the `.keytab` file to the directory you created previously and configure the container's NGINX settings to use Kerberos authentication.

In your OTOBO configuration file (`Kernel/Config.pm`), you now need to stipulate that you want to use Kerberos for authentication by copying the configuration lines that define this from the default configuration (`Defaults.pm`).

You also need to change some browser configurations for Kerberos SSO. For Chrome, Edge, and Internet Explorer, you need to add OTOBO to the local or trusted sites and enable the setting *Enable Integrated Windows Authentication*. In Firefox, you need to set `network.negotiate-auth.trusted-uris` to the OTOBO URL.

Kerberos SSO requires precisely defined settings and the correct configuration of all components. If problems occur, check the container status, the NGINX configuration, and the setup details in Kerberos and Active Directory.

Conclusions

OTOBO is a technologically advanced ticketing and help desk system. Thanks to regular development, the application is a genuine alternative to more established products. Seamless Active Directory integration, the ability to access email from Microsoft 365 by OAuth 2.0, and single sign-on exemplify OTOBO's extensive range of services and illustrate its remarkable ability to adapt to a wide variety of requirements and IT environments.

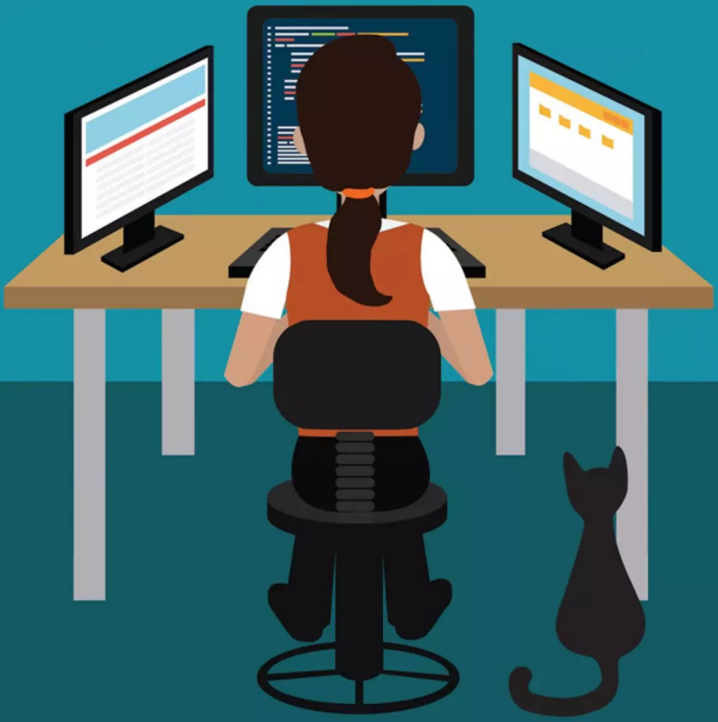
The software's open architecture is particularly worthy of praise: Existing features can be extended or reconfigured to suit your needs, while supporting integration with a wide range of third-party systems and technologies. ■

Info

[1] **OTOBO:** [https://otobo.io/en/]

[2] **System requirements:** [https://doc.otobo.org]

Want to work from anywhere?



Find remote jobs now!

OpenSource
JOB HUB

opensourcejobhub.com

@opensourcejobhub.com

Open Source JobHub

Open Source JobHub

@osjobhub@fosstodon.org

Remote access with Tactical RMM

Spy Story

The Tactical RMM open source tool offers a combination of remote monitoring and management, with agents to support Linux, macOS, and Windows. By Martin Loschwitz

The remote monitoring and management (RMM) process is slightly confusing because it takes the established term “monitoring” out of its usual context and uses it in a different way. Most administrators likely think instinctively of tools that monitor servers in the data center (e.g., Nagios or Prometheus). However, that’s not what RMM is about; instead, it focuses on an organization’s clients, monitoring them remotely with various metrics so that, if something goes wrong with a computer, the alarm bells sound for the admin in charge.

Monitoring and, in particular, managing mobile clients require different tools and processes from monitoring static servers in a data center.

In this scenario RMM providers enter the play: RMM is basically remote monitoring by an agent component with the option of remotely initiating immediate actions on the client and, where needed, being able to help resolve a problem by sharing the desktop. It is a kind of TeamViewer on steroids, if you like, with a monitoring function as the icing on the cake.

Corporate client fleets are increasingly becoming heterogeneous and mobile, so RMM tools have often become a central element of system administration. Products are available from both the open source and commercial realms. The open source candidate is Tactical RMM [1]. The program promises high performance, low maintenance overhead, and a comprehensive feature set.

It comes with agents for Windows, macOS, and Linux, covering practically all the popular common operating systems used in day-to-day business.

Paired with MeshCentral

Tactical RMM integrates seamlessly with MeshCentral [2], a web interface that supports extensive use of the tool’s feature set. However, some administrators would rightly argue that a state-of-the-art RMM tool must enable web-based administration and that Tactical RMM would be pretty useless without MeshCentral. At the end of the day, most admins are likely to use both tools together in their everyday work.

The separation between Tactical RMM and MeshCentral has practical implications for those who want to install and use the product, because you have to install and configure both Tactical RMM and MeshCentral to enjoy the full Tactical RMM experience. I look at how this partnership works, discover what you need to watch out for, and point out the stumbling blocks on the way.

Implementation as a Virtual Instance

The authors of Tactical RMM offer comprehensive documentation for their tool, illustrating several routes to a successful install. The classic approach is still that of installing on a virtual instance with a shell script downloaded from the web. This process is remarkable, because the more typical way is to deploy any required services as Docker containers. Tactical RMM also supports Docker containers but states that the legacy approach with scripting and a VM is the only officially supported route.

To get started, you first need a virtual instance with as much disk space as possible, at least four CPU cores, and no less than 4GB of

Photo by Marten Newhall on Unsplash

RAM. If your setup is not too large, you should be able to back up the environment's new data and the metrics data history for around a year on 500GB. Debian GNU/Linux 12 is strongly recommended as the technical underpinnings. If you prefer to work with Ubuntu, you can go for Ubuntu 22.04.

SSL and DNS Required

Before the installation can start, you have to meet a few requirements that mainly relate to the infrastructure Tactical RMM needs to work correctly. SSL, for example, is key during the setup process. The agents that run on the clients and need to talk to the mothership require encryption because they will be sending sensitive and often personal data subject to mandatory special protection as defined by the General Data Protection Regulation (GDPR) of the European Union. Transmitting this data across the network without encryption is unthinkable, which makes SSL more or less mandatory. This operation can only work if the server has a valid SSL certificate that the client can validate when the connection is established.

You have some very clear-cut homework. Tactical RMM cannot simply switch to a "real" certificate later if you specify the use of a self-signed certificate during installation. A valid SSL certificate for Tactical RMM must be in place from the outset of the setup process. Alternatively, Tactical RMM offers the option of automatically supplying valid certificates by Let's Encrypt, which is undoubtedly the cheapest option. For it to work, though, you need to prepare the domain in which Tactical RMM will be accessible.

More specifically, you need to make sure at least the `api`, `mesh`, and `rmm` host entries work for the respective domain. For example, assuming *example.net* is the domain name, *api.example.net*, *mesh.example.net*, and *rmm.example.net* need to point to the IP address on which Tactical RMM will be accessible. Incidentally, installing Tactical RMM on a virtual instance in a smaller office environment, which will typically reside behind a network address translation (NAT) router, is fine, but you do need to make sure you have a dynamic DNS (DDNS) entry with a sub-hosts option in place that points to the respective setup and has at least the three DNS entries referred to above. In this kind a setup, you also need to ensure that you pass HTTPS port 443 on the router to the local IP address on which Tactical RMM is running.

Firewall

Once the DNS configuration is complete, you will have a few prepping tasks for the would-be target systems on your to-do list. It is good system administrative practice to update the respective system to the latest version first with

```
apt update
apt dist-upgrade
```

Tactical RMM also needs a separate local user account, which you can create with

```
useradd -m -G sudo -s /bin/bash tactical
passwd tactical
```

The usual rules apply for the password you choose.

On Debian systems in particular, you need to install a few additional packages because they are not included in the default installation. The command

```
apt install -y wget curl sudo ufw
```

fetches all the required software. The Ubuntu firewall is entirely optional. If you want to store your firewall rules on other systems or do not want any on the system itself because a firewall sits in front of your IT landscape, you can leave out the `ufw` option. If you want to use the Ubuntu firewall, after installing the package, enter

```
ufw default deny incoming
ufw default allow outgoing
ufw allow https
```

to ensure incoming packets are always dropped and that only port 443 is open. If you also want to be able to log in to your server with SSH, you need to enter

```
ufw allow ssh
ufw enable && ufw reload
```

to enable the firewall with its updated ruleset.

Installing Tactical RMM

Working as the admin user, type

```
su - tactical
wget ⮐
  https://raw.githubusercontent.com/⮐
    amidaware/tacticalrmm/master/⮐
    install.sh
chmod +x install.sh
./install.sh
```

to become the `tactical` user, download the setup script from Tactical RMM, make it executable, and call it locally. If you want to make sure the developers are not trying to trick you, view `install.sh` in your choice of editor before you execute it. I don't say this to give you the impression that the developers are up to no good; however, many administrators now quite rightly regard as bad practice downloading a script from the Internet and executing the script without checking it first.

Immediately after launching, the installation script prompts you for a mass of details. For example, you need to specify the subdomain for the back end (*api.example.net*), for the front end (*rmm.example.net*), and for MeshCentral (*mesh.example.net*). The wizard also prompts you for the parent root domain (*example.net*) and wants you to supply an email address that will be stored on the system as a contact point for the administrator. The script then asks whether it can add local entries for the specified hosts to the `/etc/hosts` file; you will want to confirm this by pressing *y*.

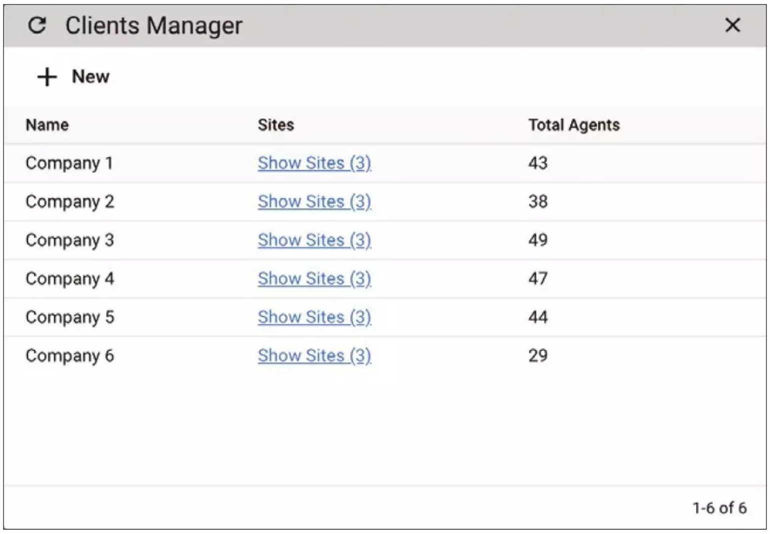
If you have decided to use Let's Encrypt, an additional intermediate step requires further intervention with your DNS service because Tactical RMM configures Let's Encrypt to use a TXT entry in the DNS to check the ownership of the (sub)domain. To let you do this, the Tactical RMM installation script shows you a matching TXT entry, which you simply need to add to your own domain's DNS configuration. Depending on your DNS provider, it can take a while for new DNS entries to propagate throughout the entire DNS system. Once the TXT entry for Let's Encrypt becomes available, the certificate you need is issued and the installation script continues without error.

Once this step has completed, the rest is very quick: The wizard prompts for a login and password for the web interface. In the next step, the login screen for MeshCentral comes up as the front end for Tactical RMM. The wizard also quits after this step; if it has not displayed an error message in the meantime, you can assume that the Tactical RMM installation including MeshCentral is working. To confirm, you can log in to MeshCentral with the username and password combination you just created.

Setting Up Tactical RMM

After the initial installation of Tactical RMM and your initial login to MeshCentral, the first task is to create a basic configuration. Internally, the tool is organized by sites and hosts – that is, by the locations and systems you want to manage (Figure 1). Correct host-to-site mappings are important because batch operations that affect all the hosts belonging to a site can be defined for a site.

What's more, Tactical RMM can have multiple tenants. In other words, the tool can manage sites belonging to several organizations, which makes it particularly practical for service providers. Anyone managing systems with RMM on behalf of their customers only needs one Tactical RMM installation, because the setup can be mapped in a meaningful way by entering the required data in Tactical RMM. To do so, first select *Settings | Clients* in the MeshCentral tenant manager and create the customers you want to manage when you get there.



Name	Sites	Total Agents
Company 1	Show Sites (3)	43
Company 2	Show Sites (3)	38
Company 3	Show Sites (3)	49
Company 4	Show Sites (3)	47
Company 5	Show Sites (3)	44
Company 6	Show Sites (3)	29

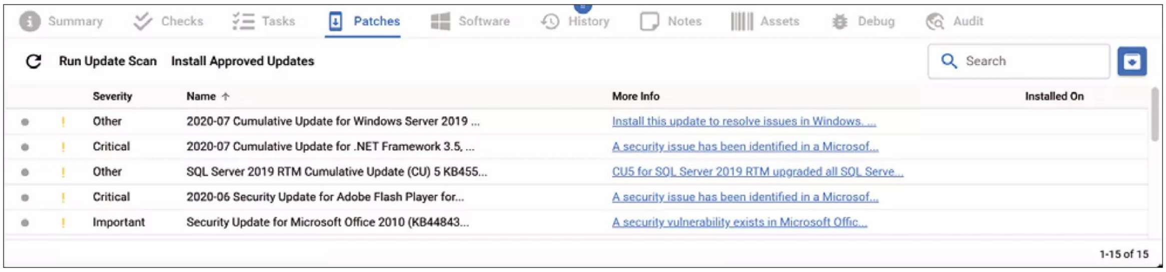
Figure 1: Tactical RMM is divided internally into sites and hosts. Sites are also assigned to clients – the tool is therefore fundamentally capable of multiple clients.

Your customers are then listed in the drop-down menu on the left. Now right-click to open the context menu with the *Add Site* item, where you can add sites for each customer. When you add agents, it also makes sense to create a *Default* site in Tactical RMM. This can be combined later with a deployment. In Tactical RMM-speak, a deployment is the list of all tasks to be carried out for a specific site belonging to an individual customer in order to deploy the agent to all systems there.

Deployments have an expiry time after which they need to be renewed. However, you can freely define the time after which a deployment expires, which will be important later when deploying the Tactical agent.

Adding Hosts

The best RMM environment is useless if it does not contain hosts that it can manage further down the line. Logically, the next step is to install the necessary agents on those systems. Again, a couple of details need to be considered, but these details are specific to the operating system.



		Severity	Name ↑	More Info	Installed On
●	!	Other	2020-07 Cumulative Update for Windows Server 2019 ...	Install this update to resolve issues in Windows. ...	
●	!	Critical	2020-07 Cumulative Update for .NET Framework 3.5, ...	A security issue has been identified in a Microsof...	
●	!	Other	SQL Server 2019 RTM Cumulative Update (CU) 5 KB455...	CU5 for SQL Server 2019 RTM upgraded all SQL Serve...	
●	!	Critical	2020-06 Security Update for Adobe Flash Player for...	A security issue has been identified in a Microsof...	
●	!	Important	Security Update for Microsoft Office 2010 (KB44843...	A security vulnerability exists in Microsoft Offic...	

Figure 2: Tactical RMM offers comprehensive reporting mechanisms, such as an overview of the patches installed on a Windows host. Actions can also be planned on the basis of this information.

If you assume the statistics regularly published on the Internet for the distribution of desktop operating systems are roughly correct, the majority of systems managed in the wild probably use Windows. On the one hand, this is good news, because the Tactical RMM Windows agent offers a number of features that are not available for the Linux and macOS agents, including, for example, the option to check the current patch status of the system remotely (**Figure 2**). On the other hand, the Tactical RMM agent setup on Windows is a little quirky.

Signed Agents Cash Only

One issue is having to cope with comprehensive antivirus (AV) protection under Windows, which is normally helpful because it ensures that the system is protected against classic viral attacks from the web. Unfortunately, antivirus software clashes with Tactical RMM because it is supposed to be able to execute its code on the system and even take control of the system for remote maintenance. From the point of view of antivirus software, Tactical RMM or its agent looks exactly like a virus.

To avoid problems, you need to stick to a couple of guidelines. First and foremost, you should use the digitally signed packages to install the Tactical agent [3]. Unlike their unsigned counterparts, however, these packages are only available if the organization running the Tactical RMM installation sponsors the project [4]. It is important to understand that signed agents are not available as part of the generic packages. Instead, Tactical RMM dynamically generates an installation package for each host you want to add and deploys this package on the target system. The digital signature is tied to sponsorship of the project.

If you need signed agents, which is true for most corporate users, you must sign up for at least the Tier 1 sponsorship level, which will set you back $55 a month; in a corporate context, this kind of expenditure will typically be manageable, especially given that, according to Tactical RMM, up to 200 target devices can be monitored in a meaningful way in this sponsorship tier.

Exceptions for Antivirus Software

Besides installing the signed agents, you also need to make sure you define an exception for the Tactical RMM agent binary in your antivirus program (i.e., you must explicitly tell your antivirus software that RMM is not a virus but is permitted to execute code on the target system). If you use

IT Highlights at a Glance

ADMINUPDATE
Issue 415 • February 5, 2025
CLOUDFEST March 17–20, 2025 Europa-Park Germany
This Week's Feature
Data Deduplication on Window
Server 2022
Data deduplication tools boost efficiency
storage management by improving stor
utilization and saving network bandwidth
backup and replication processes.
News and Resources
• Sysdig Launches Open Source Stratoshark for Cloud Observabi
• OpenVox Automation Framework Announced
• Red Hat Releases Kubernetes-Native Connectivity Link
In Case You Missed It
Integrate Remote Cloud Storage
You don't need native clients for every single service just to back up or
synchronize your data in the cloud; Rclone helps you handle these tas
multiple cloud accounts at the command line or in a graphical front en
Automated Health Checks
The open source Dradis framework facilitates creating plans for carrying out
team pentests and standardizing reports from different tools to create summary

LINUX UPDATE
No. 397 • February 6, 2025
HETZNER SCALE UP NOW! HETZNER OBJECT STORAGE BUILT TO GROW WITH YOUR NEEDS
Bringing Privacy to Social Media
The LibRedirect browser extension redirects YouTube, Instagram, Reddit, TikTok, and other services to alternative, privacy-friendly front ends.
In the News
• System76 Refreshes Meerkat Mini PC
• First Release Candidate for Linux Kernel 6.14 Now Available
In Case You Missed It
• Simplify VPN Management
WireGuard is a less complex VPN solution compared to OpenVPN and IPsec. WireGuard Easy simplifies the process even further by allowing you to operate a VPN and manage clients through a user-friendly web interface.
• Tools for Enumerating Resources
A cyberattack unfolds in stages. The enumeration phase is when the attacker looks for holes in the target system. Tools like feroxbuster and ffuf bring the power of automation to the search.

Too busy to wade through press releases and chatty tech news sites? Let us deliver the most relevant news, technical articles, and tool tips – straight to your Inbox.

ADMIN Update • Linux Update

Keep your finger on the pulse of the IT industry.

ADMIN Update: bit.ly/ADMIN-Update
Linux Update: bit.ly/Linux-Update

Photo by Andrew Neel on Unsplash

Windows Defender, the default AV tool on Windows, you can do so in PowerShell:

```
Add-MpPreference -ExclusionPath ➘
  "C:\Program Files\Mesh Agent\*"
Add-MpPreference -ExclusionPath ➘
  "C:\Program Files\TacticalAgent\*"
Add-MpPreference -ExclusionPath ➘
  "C:\ProgramData\TacticalRMM\*"
Add-MpPreference -ExclusionPath ➘
  "%TEMPDIR%\is-*.tmp\tacticalagent*"
```

Once you have completed these preparations, the rest of the agent install process is uncomplicated. Now the agent deployment for the *Default* site mentioned previously enters the scene. To set the ball rolling, click *Agents | Manage Deployments* and *New*. As a rule, you will want to keep the default settings. The new deployment will then appear in the list along with a *Copy Link* button. Copy the link and call it up in the browser on the target system.

The signed agent package then begins to download. Double-click to trigger the installation of the preconfigured agent, which automatically contacts the issuing Tactical RMM instance once the process has completed. The device user can handle this step if you do not have direct access to the device. Of course, this requires cooperation on the part of the user.

Linux and macOS Agent Drawbacks

Installing agents for Linux and macOS systems is similar to installing on Windows clients (**Figure 3**), but, as already mentioned, only the Windows agents support the full range of Tactical RMM functions. Features such as an overview of installed patches are not available for Linux or macOS.

Tactical RMM has a wide-ranging feature set. For example, the software backs up the monitoring part of RMM with comprehensive reporting. You can specify which parameters will be logged on which systems and add them to a report. Also, you can trigger both backup and recovery actions remotely. The detailed and well-written Tactical RMM documentation [5] provides a detailed overview.

It would go well beyond the scope of this article to look at all of Tactical RMM's features in detail. From an admin point of view, though, you should take a tour of the tool yourself. MeshCentral is reasonably intuitive, the menu items are clearly labeled, and the information the tools communicate is typically easily understandable.

Conclusions

Tactical RMM is a state-of-the-art tool for remote monitoring and management. The install procedure is not entirely trivial, but given appropriate preparation, it is fairly easy for an experienced admin to manage. That signed agents for Windows are only available for cash is a bitter pill to swallow – after all, secure, standardized, and compliant monitoring of Microsoft systems is not possible without signed agents. At least the fees charged by the vendor are reasonable. ■

Info

[1] **Tactical RMM:** [https://github.com/amidaware/tacticalrmm]

[2] **MeshCentral:** [https://meshcentral.com]

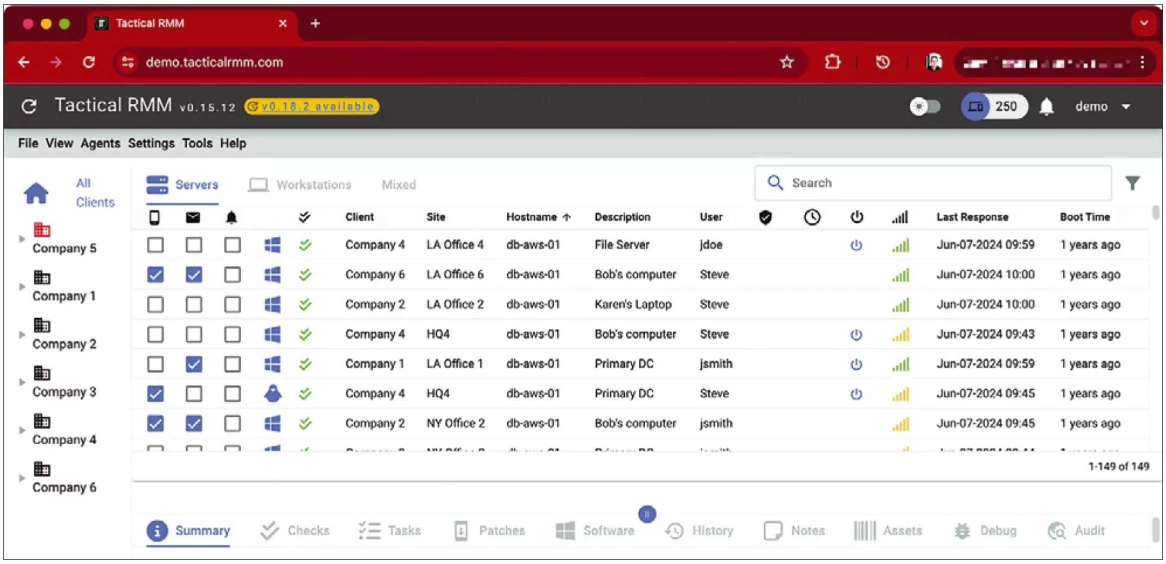
[3] **Installing agents:** [https://docs.tacticalrmm.com/install_agent/]

[4] **Code signing:** [https://docs.tacticalrmm.com/code_signing/]

[5] **Documentation:** [https://docs.tacticalrmm.com]

The Author

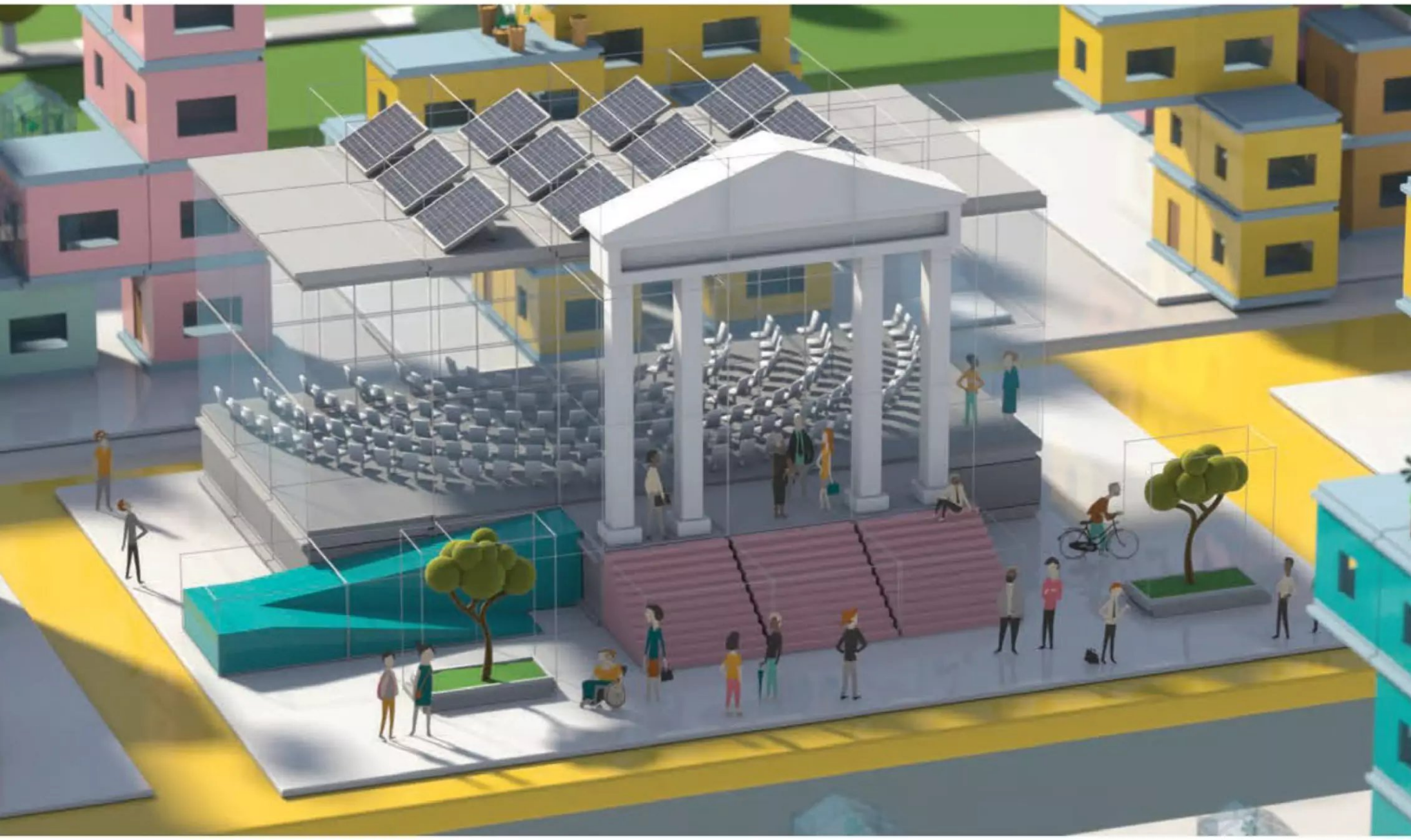
Martin Loschwitz is the founder and managing director of True West IT Services GmbH, which offers scalable IT infrastructure based on OpenStack and Kubernetes.



					Client	Site	Hostname ↑	Description	User					Last Response	Boot Time
☐	☐	☐		✓	Company 4	LA Office 4	db-aws-01	File Server	jdoe			⏻		Jun-07-2024 09:59	1 years ago
☑	☑	☐		✓	Company 6	LA Office 6	db-aws-01	Bob's computer	Steve					Jun-07-2024 10:00	1 years ago
☐	☐	☐		✓	Company 2	LA Office 2	db-aws-01	Karen's Laptop	Steve					Jun-07-2024 10:00	1 years ago
☐	☐	☐		✓	Company 4	HQ4	db-aws-01	Bob's computer	Steve			⏻		Jun-07-2024 09:43	1 years ago
☐	☑	☐		✓	Company 1	LA Office 1	db-aws-01	Primary DC	jsmith			⏻		Jun-07-2024 09:59	1 years ago
☑	☐	☐		✓	Company 4	HQ4	db-aws-01	Primary DC	Steve			⏻		Jun-07-2024 09:45	1 years ago
☑	☑	☐		✓	Company 2	NY Office 2	db-aws-01	Bob's computer	jsmith					Jun-07-2024 09:45	1 years ago

Figure 3: **Besides Windows, Tactical RMM also supports macOS and Linux as target operating systems, as indicated by the operating system icons under Workstations in the overview.**

Public Money
Public Code



Modernising Public Infrastructure with Free Software

fsfe Free Software Foundation Europe

Learn More: https://publiccode.eu/

Centrally manage configuration files

Stow It!

The GNU Stow tool in combination with Ansible can help you manage configuration files. By Thorsten Scherf

Individual application configurations, saved in what are known as dotfiles, are hidden files typically kept in a user's home directory. You can manage these files centrally in a number of ways. Applications usually search for their configuration either in a user's home directory or in `~/.config`, which is the default for the `XDG_ CONFIG_HOME` variable. It is part of the Cross-Desktop Group's (XDG's) generic XDG base directory specification [1], which includes several standards for the configuration of desktop applications and also applies to text-only applications. Of course, you'd be smart not to rely on your own configuration files being available simply locally but to store them centrally, as well (e.g., in a Git repository). A relatively easy approach would be to store all the dotfiles in a central directory under Git's control and then simply define symlinks to the files. However, this method will not do the trick, because the applications themselves also need to be installed there, including all their dependencies and the plugins they use, as expected by the configurations.

GNU Stow and Ansible

A very simple but powerful solution to this problem involves the use of just two tools: Stow and Ansible. Stow [2] comes from the GNU project and was originally developed to manage manually compiled software with symlinks. One nice side effect is that you can apply this capability to configuration files.

Stow manages related files in so-called packages, which are located in a Stow directory. The target folder is the directory in which Stow creates symbolic links to the packages and the files they contain and is typically a user's home directory, the `~/.config` folder, or the folder assigned to the `XDG_CONFIG_HOME` variable. In other words, if you use `~/dotfiles` as your Stow directory, and it contains the package *zsh* with the zshrc file, you would call Stow as follows to generate a symbolic link in your home directory:

```
stow --target $HOME ➘
     --dir $HOME/dotfiles/ zsh
```

The results are now:

```
ls ~/.zshrc
/root/.zshrc -> dotfiles/zsh/.zshrc
```

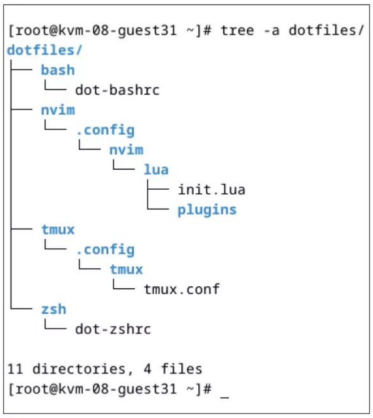
Figure 1 shows a simple Stow directory structure. Because the configuration files are often hidden, Stow offers a really neat feature for working with more than just empty package directories. Instead of the dot at the beginning of the file name, you can simply use `dot-`. If you then call Stow with the `--dotfiles` option in place,

```
ls ~/dotfiles/zsh/dot-zshrc
/root/dotfiles/zsh/dot-zshrc

stow --dotfiles --target $HOME ➘
     --dir $HOME/dotfiles/ zsh

ls -l ~/.zshrc
/root/.zshrc -> dotfiles/zsh/dot-zshr
```

the tool turns `dot-` back into a dot and hides the file in the target directory.



```
[root@kvm-08-guest31 ~]# tree -a dotfiles/
dotfiles/
├── bash
│   └── dot-bashrc
├── nvim
│   └── .config
│       └── nvim
│           └── lua
│               ├── init.lua
│               └── plugins
├── tmux
│   └── .config
│       └── tmux
│           └── tmux.conf
└── zsh
    └── dot-zshrc

11 directories, 4 files
[root@kvm-08-guest31 ~]# _
```

Figure 1: The Stow directory contains packages that, in turn, comprise configuration files.

Photo by The New York Public Library on Unsplash

Ansible Role and Playbook

To install the individual software packages and the matching dependencies, you need to create a new Ansible role. In this role, you can then also call the `stow` tool to deploy the configuration files to the desired system. To create a new role with `ansible-galaxy`, for example, use:

```
ansible-galaxy role init dotfiles
```

The tool creates the required directory structure with the files you need to develop a new role for dotfile management:

```
ls dotfiles/
defaults   files  handlers   meta
README.md  tasks  templates  tests
vars
```

To set some default variables in the `defaults/mains.yml` file, so you can access them within the task file, add the lines:

```
---
dotfiles_repo: ➘
  "https://git.example.com/➘
   users/tscherf/dotfiles.git/"
dotfiles_repo_local_destination: ➘
  "{{ ansible_env.HOME }}/dotfiles"
dotfiles_home: "{{ ansible_env.HOME }}"
```

At the very least, you will want specify the Stow directory and target directory and the URL repository for your dotfiles here.

Defining Tasks

Listing 1 contains some simple examples of tasks that this Ansible role will now execute. It only uses modules that are part of the *ansible-core* package and are therefore available on every system on which Ansible is installed. The package manager module used here is `dnf`, but of course you can also use a different module, depending on your choice of package manager for installing software.

Besides installing the packages Stow and Git, the applications for which you are providing configuration files are also installed. A task then handles checking out from the Git repository. Afterward, a list of the individual Stow packages in this repository is generated. As the last task, you then use this list and Stow to link the files of the individual packages from the target directory.

Ansible Playbook

To deploy this role on a system, you can use the playbook:

```
---
- hosts: localhost
  roles:
    - { role: dotfiles }
  become: true
```

This playbook is written with the assumption that it will be deployed on the system on which you will be using the Ansible role. As with the role itself, you can adapt the playbook, for example,

```
ansible-playbook dotfiles-play.yml
```

to suit your own requirements before calling it up.

Conclusions

The advantage of storing configuration files centrally is that they are always available, even on new systems. The combination of Ansible and Stow offers an easy approach to managing configuration files, and you can install software packages and plugins for your applications, as well. If you want to find out more about Ansible roles, the Ansible documentation [3] is a good place to start.

Listing 1: Tasks in `main.yml`

```
---
- name: Update dnf cache
  ansible.builtin.dnf:
    update_cache: true
  become: true

- name: Install stow and git
  ansible.builtin.package:
    name:
      - stow
      - git
    state: present
  become: true

- name: Install terminal applications
  ansible.builtin.package:
    name:
      - neovim
      - tmux
      - zsh
    state: present
  become: true

- name: Ensure dotfiles repository is cloned on target
       system
  ansible.builtin.git:
    repo: "{{ dotfiles_repo }}"
    dest: "{{ dotfiles_repo_local_destination }}"
  become: false

- name: Build package list
  ansible.builtin.find:
    paths: ["{{ dotfiles_repo_local_destination }}"]
    depth: 1
    file_type: directory
  register: files
  become: false

- name: Deploy dotfiles
  with_items: '{{ files.files }}'
  ansible.builtin.command:
    cmd: stow --dotfiles --target=
      {{ dotfiles_home }} --dir
      {{ dotfiles_repo_local_destination }}
      {{ item.path | basename }}
    chdir:
      "{{ dotfiles_repo_local_destination }}"
    register: result
    changed_when: 'result.stderr is
    search("LINK: ")'
  become: false
```

Info

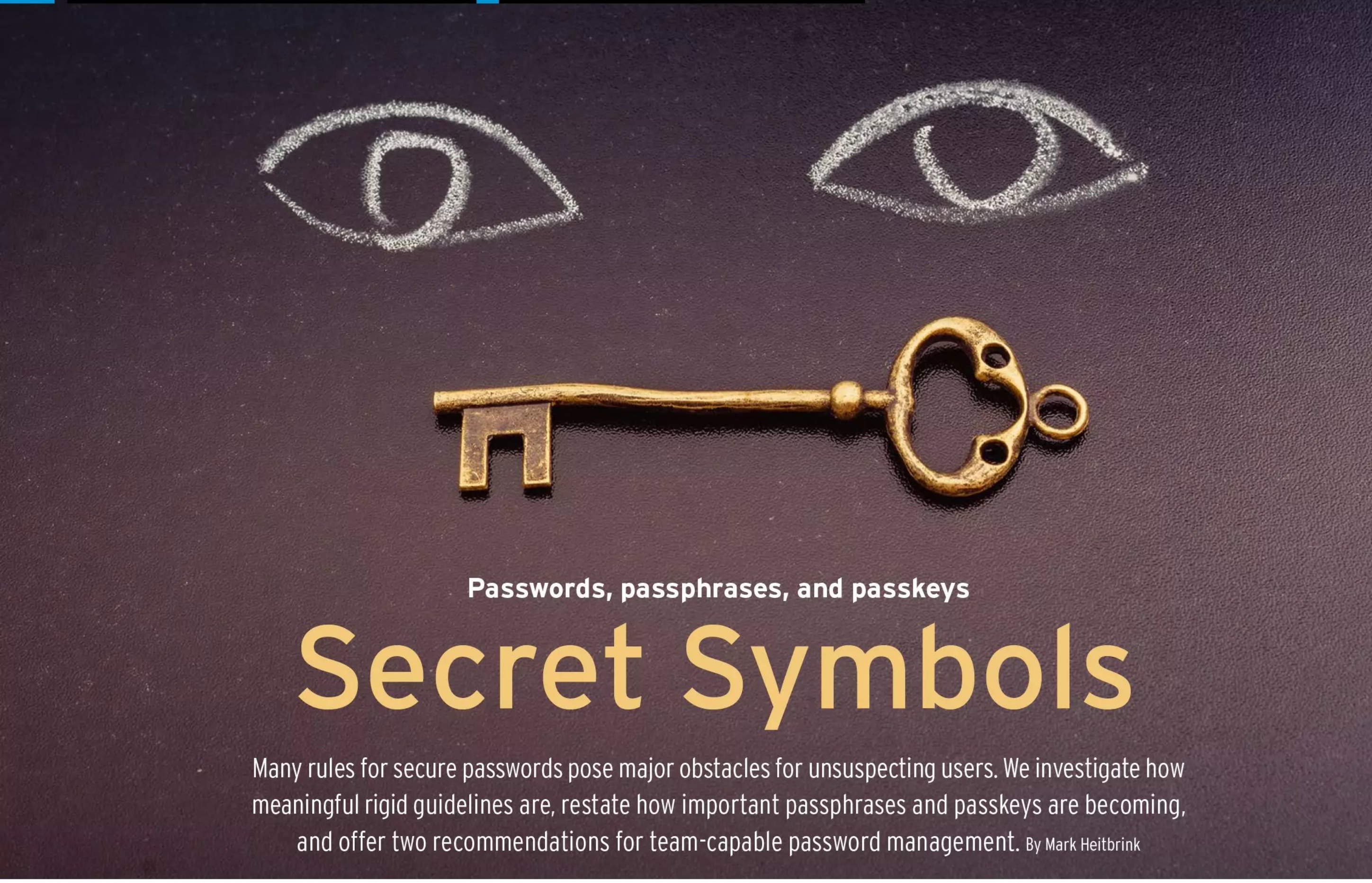
[1] XDG base directory specification: [https://specifications.freedesktop.org/basedir-spec/latest/]

[2] Stow: [https://www.gnu.org/software/stow/]

[3] Ansible role documentation: [https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_reuse_roles.html]

The Author

Thorsten Scherf is the global Product Lead for Identity Management and Platform Security in Red Hat's Product Operations group. He is a regular speaker at various international conferences and writes a lot about open source software.



Passwords, passphrases, and passkeys

Secret Symbols

Many rules for secure passwords pose major obstacles for unsuspecting users. We investigate how meaningful rigid guidelines are, restate how important passphrases and passkeys are becoming, and offer two recommendations for team-capable password management.

By Mark Heitbrink

The classic password has a long history and the terms, along with the times, are a-changing. At the end of the day, everyone still uses passwords, but the term passphrase, for example, describes a different way of creating and remembering them. After all, every attempt to access something starts with a secret that has to be resolved and handed over. One part of the puzzle is the username, which identifies the owner. Under normal circumstances, the owner should be the only person who has the key to access. When generating a password, the user faces two challenges: the need to generate a secure password, and the need to remember it.

Unrealistic Rulesets

Bill Burr wrote the official password guide for the U.S. National Institute of Standards and Technology (NIST). According to the guide, a password should comprise at least eight characters and be complex. It should contain upper- and lowercase letters, numbers, and nonstandard characters (i.e., three to four different types of characters ideally), and it should be changed regularly. Ideally, a password history will prevent previously used passwords being rehashed, or at least enforce a long wait before reuse.

Unfortunately, many things in this ruleset have been misinterpreted and, above all, users have found unplanned or unintended detours and shortcuts. When it comes to the length of the password, for example, the emphasis is on “at least” eight characters. This has led many users to view these eight characters as the maximum, and in many cases, IT departments fail to communicate the requirements clearly. Of course, a password can and should have more than eight characters.

In practice, passwords created according to the above-mentioned rules (e.g., *dU7%9IL&*) that you are then forced to change every three months will quickly mutate to *Summer25*, because it is complex according to the rules. Three out of four character types (check), eight characters (check) – the requirements are met. From this logic, you probably can guess which password will be used in three months’ time. Users will make it as easy as possible for themselves, because they just want to work; what they definitely don’t want is to be forced to type a complex password every 15 minutes when their device is locked by the screen saver or the inactivity limit.

The idea of using a keyword phrase or the first letters of a song lyric or refrain, perhaps including a letter swap (e.g., A becomes 4, E becomes 3, O becomes 0, 1 becomes !, etc.), makes the password even more complicated and does not make it any easier to handle. For example, the Robbie Williams refrain *Hell is gone and heaven’s here* could become *RW-H1gahh*. It has upper- and lowercase letters, a nonstandard dash character, and the number 1 instead of the letter I. The password is complex and secure according to the known rules. The only problem is that you are unlikely to remember it and will need considerably more time to remember how the abbreviation worked.

The dilemma is that as soon as the process becomes a pain and requires energy, the human brain switches off and, thanks to evolution, looks for relief and a simpler construct. In the end, the passwords are discovered on Post-it notes or scraps of paper under the keyboard, or they end with

Lead Image © berkay08, 123RF.com

something trivial, such as the season, an astronomical object, the current month, the current year, or another counter that simply increments (e.g., *Michael01*, *Michael02*, *Michael03*, etc.).

Passphrases Make It Easy

The aim therefore must be to have a password that is easy to remember; although remembering it should not be difficult, the password should still be secure. This path arrives at passphrases, which comprise three, four, or five words that have no mutual associations. A separator character is easily added, and the question of whether or not to add a number is at the user's discretion – any worries about this construction is addressed later.

With the use of German word list 7776 from dys2p [1] (saved as a TXT file), I created a phrase with three or more words in a simple PowerShell custom passphrase generator:

```
$Count=Read-host "Number of words"
$Separator=Read-Host "Separator"
$Content=Get-Content ➘
  -Path "$PSScriptRoot\Wordlist.txt"
$Phrase=Get-Random $Content -Count $Count
$Phrase -join $Separator
```

The result is phrases like *Anprangern-Verknallen-Erkunden* (denounce-crush-explore) or *Jawort#Einwurf#Imbiss* (consent#insertion#snack). At the end of the process, you have three simple words with a unique sound and an easy-to-remember sequence. Of course, you could also look around the room and use the names of things in your field of view, such as *lamp-poster-window*. The aim is always to make things as easy as possible for users.

Even Microsoft relies on the option of passphrases for the Local Administrator Password Solution (LAPS) in Windows Server 2025 [2]. Passphrases are easier to type, easier to picture in your mind, and definitely easier to remember; they are humanly readable, so you do not need to type them out painstakingly character by character, but just enter them easily word by word.

This ease of use makes a passphrase your first and most important line of defense before accessing a system. The password must not be trivially guessable. The code snippet above can be used in a website on the intranet published for your users. The practical thing about this solution is that it automatically fulfills the most important rule when dealing with passwords and code phrases: length.

Length Counts

The password length is crucial for attacks based on trial and error (brute force). The longer the password, the longer the compute time required to run through the list of passwords.

At this point, you come back to the concerns mentioned above; however, length always beats complexity.

A nonstandard character automatically gives you a higher multiplier in the formula, whereas each additional character increases the power, which results in an incredible number of possibilities for each additional character. For example, *denounce-crush-explore* has 22 characters. From the current state of the art and calculations by Hive Systems [3], you must assume that all passwords with fewer than 16 characters are no longer secure.

The difficulty now is to sell this to the user in a psychologically effective way. If you send a message to staff with the header "Minimum 16-character passwords starting tomorrow," your users are guaranteed to be up in arms and asking management if they can have some of the stuff the people in IT smoke. If you choose instead a header such as "Simple passwords that you can keep forever, starting tomorrow" and then offer the passphrase generator as a web front end, users will not even realize you are getting them to do something they were terrified of doing.

The script could be modified to use two words, and you could include a minimum of *x* characters. The only important thing to users is a quick and easy way to create passwords.

A long password can be memorized by repeatedly typing it and is likely to remain as a solid image in the users' minds given time. Mistyping, even with 25 characters or more, is becoming increasingly rare, all of which contributes toward burying the problem of trivial passwords and incremental counters.

Mandatory 2FA and MFA

Passphrases only offer basic protection and leave two unresolved problems. The password is the only protection for access, and even a 31-character passphrase could become known and be exploited at some point; moreover, many users use a password in various services on the Internet and make it their new standard. If an email address and password combination is stolen from a web store or database, it will end up in the password lists of attackers. Users who fail to comply with the ban on password recycling can expect major loss and damage.

A potential solution is not in the hands of the user. It involves combining the password with a dynamically generated key. The admins of the system that the user wants to access need to support a two-factor authentication (2FA), multifactor authentication (MFA), or one-time password (OTP) option. Knowledge of the secret alone is then no longer sufficient to gain access.

The user needs to generate another key with an additional device. In most cases, this key is a six-digit numerical code that is updated every 30 seconds. During setup, the software that displays this code (e.g., on a smartphone) is linked to the company's own technology once only.

The two big players in this field are Microsoft Authenticator and Google Authenticator. Authy has also been used, but unfortunately, this service recently suffered the loss of 33 million user datasets. The attackers are said not to have gained access or stolen any keys, but they do have millions of valid datasets – a wave of phishing may be just over the horizon.

Besides these apps, some authentication providers also offer a procedure involving SMS and email. In this case, the codes are valid for a longer period of time. Although convenient, you should not use push confirmation as an option here, if possible, because the user only has to confirm that they are currently logging in. Practice has shown that the user is once again the weakest link in the chain. Users tend to choose a weak, guessable password and feel secure because of 2FA or MFA.

Once an attacker has gained knowledge of the password by guessing or testing, they will try to log in time and time again. The device with the app then responds and constantly reports the attempt, prompting the user to say whether or not it is valid. At 2am, when the user just wants to sleep, they are likely to confirm eventually just to get some peace.

This annoying push notification attack is not in itself an attack on the construct as such, which can still be considered secure. Instead, it is in the same league as any other social engineering attack. Despite this risk of the weakest link in the chain being exploited, 2FA, MFA, and OTP remain an essential protection factor, and you simply have no excuse to leave it out. Every current cybersecurity insurance company considers it grossly negligent if external access is set up without a second factor and will not pay out in the event of data loss.

Passkey Remains on the Device

Another security measure that is also not in the hands of the user is the change from a password or passphrase to a passkey **[4]**, which is similar to a smart card and is basically login by certificate. This option has been available to Windows admins for 25 years for authentication against Active Directory but has never achieved widespread use. The reasons lie in the clumsy infrastructure that admins need to create and maintain.

The same caveat applies here: A public key infrastructure in the organization that issues certificates for users, with which they then authenticate, must never fall into the wrong hands. The handling of smart cards (plastic cards or USB sticks) has proven to be problematic in practice. The solution is smartphones, which everyone has today. An infinite number of certificates (passkeys) can be stored on the phone and assigned to a service. Another benefit is that the phone is not kept next to the computer but is carried by the user. The smartphone itself is protected against access by a PIN or biometrics (e.g., facial recognition, fingerprint ID). Depending on the application, in-app authentication might also be required. The company or external provider provides the infrastructure and is responsible for ensuring security. The user only has to deal with the end product stored in an app.

The biggest difference is private and public key encryption; instead of a password, which has to be sent to the server every time, the most important part of the secret, the private key, remains on the device and never crosses the ether or the wire. Identification is based on a signature, and that can only be generated with the private key.

The Right Password Manager

In reality, you are likely to remain trapped in the password universe for the foreseeable future, which means you have to rely on tools that help you handle passwords securely. One indispensable tool in the password universe is the password manager: This piece of software helps you generate an individual password for any website and any access point. The password is then stored in a database, which is secured by a master password and encrypted. A few important points to consider when choosing a suitable password manager include:

- The database must be located on-premises or in your own cloud. Under no circumstances should it be located with a provider behind an access portal that you do not control.
- If you opt for a cloud provider, the provider must not offer an option to restore the data if access is lost. Backup and restore must remain in your own hands. Access must be restricted to administrators who work for your own organization. If outside access were allowed, the setup would be vulnerable to attack and no longer trustworthy.
- Access to the database, if possible over the Internet, must always be secured by 2FA, MFA, or a passkey.
- Open source is desirable. The hope that people without a financial claim will keep an eye on the code, and maybe even improve it, might be a fallacy, but it is preferable to relying on a commercial provider running proprietary code that cannot be reviewed.
- The password manager must be able to generate passphrases: A 60-character zoo is massively secure but difficult to type on a device without a 105-key keyboard. Some nonstandard characters are almost impossible to type on a mobile device or are not available at all.
- The user-friendliness (i.e., the interface) of the software must follow modern style guides and be responsive when a website is provided.

The open source KeePassXC **[5]** is one of the best-known tools in the password manager world. It offers everything a non-business user could desire, which is unfortunately also the knock-out criterion for corporate use because it is a personal database without the option of sharing access. In a corporate environment, however, supplier websites, web stores, and other services need to be available to several people. KeePassXC does not offer an out-of-the-box solution in such cases.

The tool is top of the list for private use, however. KeePassXC feels at home on any operating system; it provides a database and lets you encrypt the information stored in it. A well-thought-out user interface makes KeePassXC easy to use. It is modern and offers the same look on any platform. Plugins for Chrome and Firefox communicate directly with the database. The software is available for installation, along with a portable edition, and it supports the Windows command-line package managers WinGet (also called the Windows Package Manager) and Chocolatey. As a standalone tool for anyone who simply wants to manage their own passwords, KeePassXC is an impressive solution with a mature feature set.

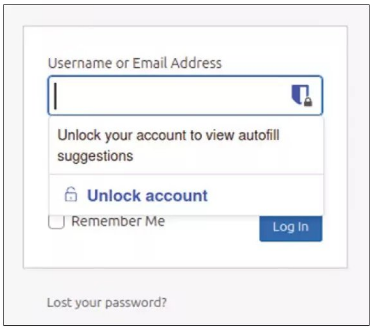
Password Manager with Team Function

As soon as the scenario switches to team functionality, with a focus on corporate use, it is often difficult to find arguments in favor of free open source tools. Organizations are often reluctant to use a tool for which they cannot officially purchase support. That said, I'll look at two free community offshoots that address this need and are also officially hosted and supported products from their providers.

Bitwarden [6] has become widespread in recent years. The tool offers a free platform for private individuals and has a manageable cost framework for families. Companies can opt for a cloud or self-hosted version. In addition to the commercial version, the open source offshoot Vaultwarden [7], which provides the full set of enterprise features hosted by Bitwarden, can be obtained on GitHub as a Docker container.

For admins, one of Bitwarden's most interesting features – besides its capability as an enterprise-grade password manager – is the ability to use policies to control deployment [8]. The endpoints or server names of a local environment can be distributed by registry entries or JSON. Bitwarden can also synchronize with Active Directory, Entra ID, Google, and Okta to discover the user accounts stored there.

Bitwarden also provides a browser plugin that can be used with web forms (Figure 1) and can access the database directly. If a suitable entry is identified by URL, you can enter the stored information directly (Figure 2). For security reasons, however, you do not have a completely automatic form fill: Choosing and storing the password has to remain firmly under the control of the user; otherwise, it would be all too easy for data to end



Unlock your account to view autofill suggestions

Figure 1: Integration of Vaultwarden or Bitwarden in a web form before unlocking access.

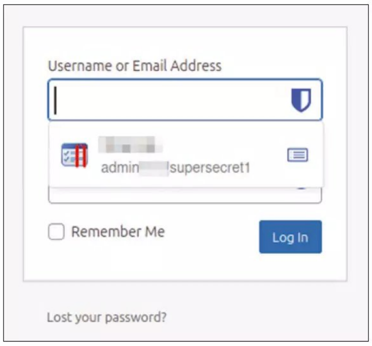
SHOP THE SHOP
shop.linuxnewmedia.com/shop

GET PRODUCTIVE WITH
COOL LINUX HACKS

Improve your Linux skills with this cool collection of inspirational tricks and shortcuts for Linux geeks.

- Google on the Command Line
- OpenSnitch Application Firewall
- Parse the systemd journal
- Control Git with lazygit
- Run Old DOS Games with DOSBox
- And more!

ORDER ONLINE:
shop.linuxnewmedia.com/shop



[redacted]
admin[redacted]supersecret1

Figure 2: If the URL matches, Vaultwarden and Bitwarden make a suggestion for the appropriate credentials.

up in the wrong form by mistake and for sensitive information to be lost.

If you are worried about free software – or you are not allowed to use it – and prefer a commercial product, you might want take a look at Passbolt [9]. The tool is practically in the same league as Bitwarden. The community edition is marketed under the same name, but it lacks a few features, such as LDAP synchronization. Each organization must decide for itself which tool is the best fit for its own needs. KeePassXC is ruled out if you need password sharing and group functionality. At the end of the day, the decision between the two free versions of Vaultwarden and Passbolt will probably come down to the graphical user interface or the desktop client. I recommend testing and evaluating the two systems. The instructions for installing the Docker-based variant are very easy to understand, and you only need minimal knowledge of Linux.

Of course, secure operation of a password server also requires familiarity with the basic feature set; if needed, some service providers offer support. For an initial test run, you could even set up a password server on a Raspberry Pi as part of a DIY project (Figure 3). After completing trials, you can then install the system on more robust underpinnings and provide it in a secure environment that includes backup and restore.

Conclusions

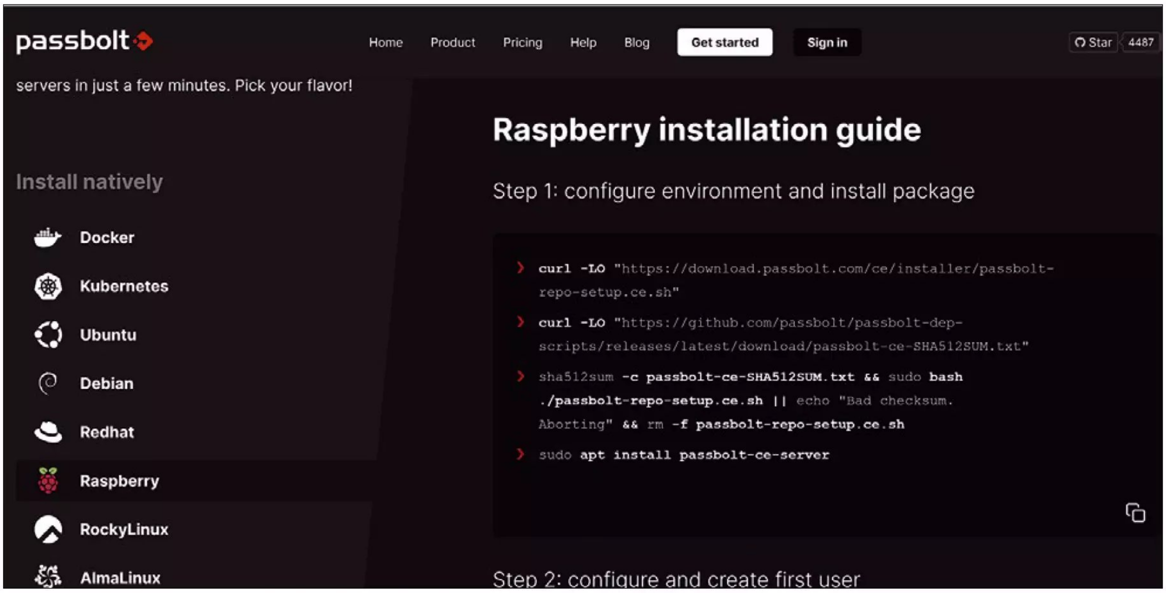
Passwords will be around for a very long time. The faster the current crop of graphics cards computes, the more important password length becomes. The last few years have seen rapid growth rates in brute-force scenarios. Password length is the only factor that the user or the organization itself can control. Additional safeguards, such as the aforementioned two-factor or multifactor authentication – or even a complete shift to passkeys – might be in the hands of the IT department but are still not supported by all service providers. Organizations need to point out security deficiencies as such and insist on the gaps being plugged. ■

Info

[1] Word list 7776 from dys2p: [https://github.com/dys2p/wordlists-de/blob/main/de-7776-v1.txt]
[2] Passwords and passphrases for Windows LAPS: [https://learn.microsoft.com/en-us/windows-server/identity/laps/laps-concepts-passwords-passphrases]
[3] Password length table from Hive Systems: [https://www.hivesystems.com/blog/are-your-passwords-in-the-green?utm_source=header]
[4] Passkeys.io: [https://www.passkeys.io]
[5] KeePassXC: [https://keepassxc.org]
[6] Bitwarden: [https://bitwarden.com]
[7] Vaultwarden: [https://github.com/dani-garcia/vaultwarden]
[8] Bitwarden deployment: [https://bitwarden.com/help/configure-clients-selfhost/]
[9] Passbolt: [https://www.passbolt.com]

The Author

Mark Heitbrink has worked in IT since the early 1990s. He's been a consultant, trainer, speaker, and author covering topics such as Active Directory, Group Policy, deployment, and public key infrastructure. Mark's current focus is mainly on security and privacy for protection against ransomware. He resides in Regensburg, Bavaria.



Raspberry installation guide

Step 1: configure environment and install package

```
❯ curl -LO "https://download.passbolt.com/ce/installer/passbolt-
  repo-setup.ce.sh"
❯ curl -LO "https://github.com/passbolt/passbolt-dep-
  scripts/releases/latest/download/passbolt-ce-SHA512SUM.txt"
❯ sha512sum -c passbolt-ce-SHA512SUM.txt && sudo bash
  ./passbolt-repo-setup.ce.sh || echo "Bad checksum.
  Aborting" && rm -f passbolt-repo-setup.ce.sh
❯ sudo apt install passbolt-ce-server
```

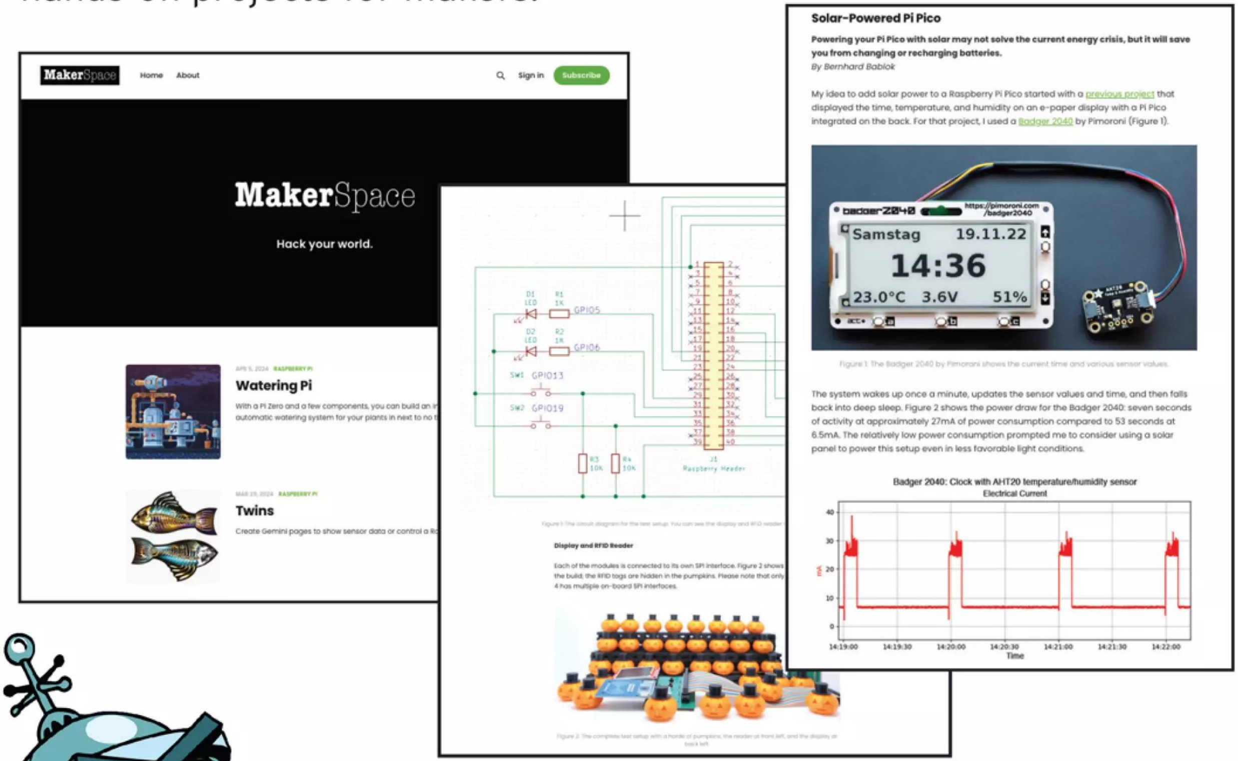
Step 2: configure and create first user

Figure 3: Passbolt comes with instructions on how to set up a password server on a Raspberry Pi.

MakerSpace-Online

New Maker Content Every Week

At MakerSpace, we are all about technology you can use to build your own stuff. Our goal is to help you turn your ideas into reality with hands-on projects for makers.



MakerSpace Home About Sign in Subscribe

MakerSpace

Hack your world.

APR 5, 2024 RASPBERRY PI

Watering Pi

With a Pi Zero and a few components, you can build an i
automatic watering system for your plants in next to no t

MAR 29, 2024 RASPBERRY PI

Twins

Create Gemini pages to show sensor data or control a R

Display and RFID Reader

Each of the modules is connected to its own SPI interface. Figure 2 shows
the build; the RFID tags are hidden in the pumpkins. Please note that only
4 has multiple on-board SPI interfaces.

Solar-Powered Pi Pico

Powering your Pi Pico with solar may not solve the current energy crisis, but it will save you from changing or recharging batteries.

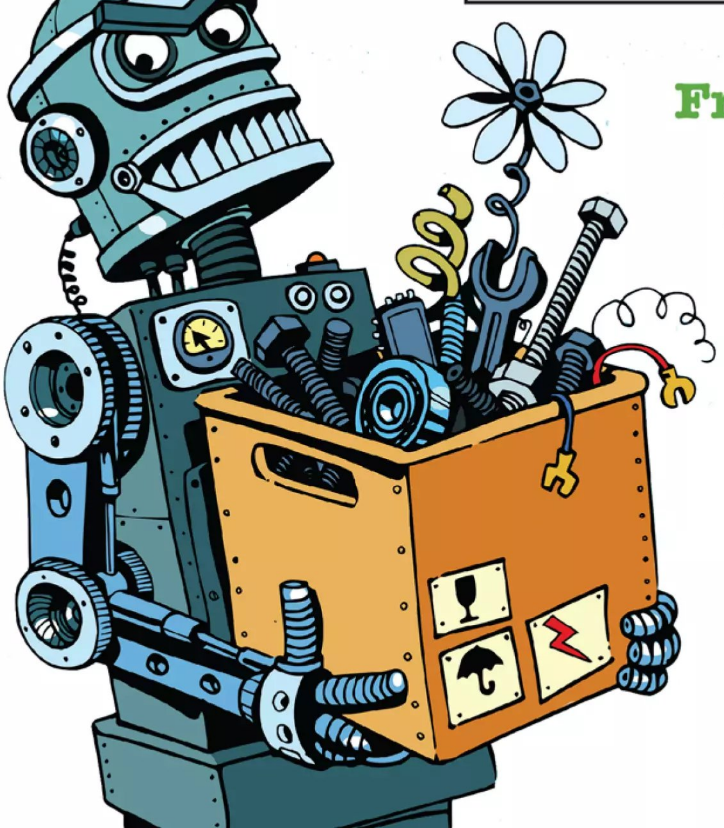
By Bernhard Bablok

My idea to add solar power to a Raspberry Pi Pico started with a previous project that displayed the time, temperature, and humidity on an e-paper display with a Pi Pico integrated on the back. For that project, I used a Badger 2040 by Pimoroni (Figure 1).

Figure 1: The Badger 2040 by Pimoroni shows the current time and various sensor values.

The system wakes up once a minute, updates the sensor values and time, and then falls back into deep sleep. Figure 2 shows the power draw for the Badger 2040: seven seconds of activity at approximately 27mA of power consumption compared to 53 seconds at 6.5mA. The relatively low power consumption prompted me to consider using a solar panel to power this setup even in less favorable light conditions.

D1 LED
R1 1K
GPIO5
D2 LED
R2 1K
GPIO6
SW1 GPIO13
SW2 GPIO19
R3 10K
R4 10K
J1
Raspberry Header
badger2040
https://pimoroni.com /badger2040
Samstag 19.11.22
14:36
23.0°C 3.6V 51%
Badger 2040: Clock with AHT20 temperature/humidity sensor
Electrical Current
mA
40
30
20
10
0
14:19:00
14:19:30
14:20:00
14:20:30
14:21:00
14:21:30
14:22:00
Time



Fresh Content, Delivered

Subscribe now and join the MakerSpace community. You'll stay up-to-date when new content is published.

Join Now!

https://makerspace-online.com

Designing a secure Active Directory

Toughen Up!

Many experienced administrators find that Active Directory is particularly problematic in terms of security. We reflect on the secure design, rigorous hardening, and security-conscious management of this service.

By Evgenij Smirnov

Active Directory (AD) does not have a good reputation among many IT security specialists, and more than one admin thinks it should be removed as soon as possible. Of course, survivorship bias plays a role in such thoughts – infrastructures that require high-caliber specialists to rescue the day are ultimately those where the greatest discrepancy between lure for attackers and the degree of hardening exists. In this article, I use the term “hardening” to refer both to IT infrastructures themselves and to the people and processes involved in managing them.

Active Directory can be run in a pretty much secure way if the organization involved is at least prepared to drop old habits and invest in a secure, state-of-the-art design instead of just in third-party tools. This way might not sound as sexy in the annual report as the latest extended detection and response (XDR), managed detection and response (MDR), or identity threat detection and response (ITDR) strategies, but it is likely to offer greater benefits.

When AD Is Not AD

To get the ball rolling, the term “insecure AD” typically means Active Directory Domain Services (ADDS) – that is, a directory service tied to Kerberos authentication and group policies. In some cases, the Active Directory Certificate Services (ADCS), Microsoft’s implementation of a public key infrastructure (PKI), deals the final death blow to the security of the environment.

However, three other ADs belong to the Windows server family. First is the Active Directory Rights Management Service (ADRMS), a cryptography tool for protecting digital content such as documents, email, etc. The service is very rarely used in local environments but provides the technology basis for Azure Information Protection, which evolved into Microsoft Information Protection and finally Microsoft Purview Information Protection [1]. As a rule, ADRMS has no influence at all on the security of the infrastructure, apart from ingenious scenarios in which a stolen message is used for spear phishing to obtain privileged credentials.

Second, you need to look at the Active Directory Lightweight Directory Services (ADLDS) – formerly known as Active Directory Application Mode (ADAM) – which is an LDAP-compliant directory service that builds on ADDS multimaster replication mechanisms but does not support computer memberships, Kerberos, and group policies. Instead, the schema is freely configurable and can be used in scenarios for which ADDS is completely unsuitable. Additionally, the service can listen on any port and multiple instances of LDS can run side by side on the same Windows server. This server role can certainly enhance the security of an ADDS organization – for example, by enabling the creation of a metadirectory by IT managers for the DMZ and other potentially insecure scenarios and to avoid direct AD queries from insecure zones.

Third is the Active Directory Federation Services (ADFS), a Security Assertion Markup Language (SAML) and OpenID Connect-compliant identity provider (IdP) for federated authentication. In conjunction with applications that support SAML authentication (these are primarily applications with a web front end), a well-configured ADFS farm can significantly improve the security of the underlying AD. On the other hand, a poorly configured ADFS farm opens up completely new attack vectors and makes securing the entire environment all the more difficult. The situation becomes particularly precarious when ADFS is not only used on the enterprise LAN, but for external applications, as well (e.g., in conjunction with Entra ID).

Known Security Issues

Active Directory’s bad reputation is not attributable to specific circumstances or players. Instead, it is a team effort: Microsoft set things off with design decisions that permanently dictated a specific development path. Above all, though, you need to look at default settings, which

Photo by Samantha Sophia on Unsplash

were already outdated at the time of their inception but had to kowtow to the manufacturer's philosophy at the time, according to which backward compatibility and user convenience took priority over all performance features and, more frighteningly, over security.

On the basis of Microsoft technology, architects, consultants, and book authors then wrote certain "design patterns" in stone, defining best practices for decades, although the world just kept on turning and Windows, AD, and the infrastructure underneath evolved with security aspects becoming increasingly important. Also, people who operate AD infrastructures contribute to insecure and unreliable everyday operations with their shortcuts, convenience practices, outdated tools, and bad scripting habits.

Although it is impossible to change the characteristics of the technology itself, it is important to be aware of them because they constitute the immovable framework of AD design and operation. The most important aspects that guide design decisions from a security perspective include:

- Write and management permissions in AD assigned to principals from AD itself. This is the technological basis for all privilege escalation attacks, including hijacking the entire AD forest.
- The difficult role of a domain controller (DC). It has its own machine identity but no security database of its own. It can replicate, but so can others.
- Replication mechanisms. Although each DC has its own copy of the AD domain, it can only work correctly if it knows that its copy is up to date and matches those of the other DCs in the domain. This complex situation opens hundreds of attack vectors, from denial of service to hijacking of the entire forest, or even neighboring forests.
- The ability to self-register in the AD-integrated DNS. This process is not tied to consistency checks and opens vectors for all kinds of name spoofing, which in turn is essential for man-in-the-middle attacks.
- Identical cryptography for NT LAN Manager (NTLM) and Kerberos RC4. The reason lies with the migration of NT domains to AD, the negative effects of which (e.g., the overpass-the-hash attack) are serious.

On the architecture side, you will come across some classics time and time again in daily practice. Many of these design anti-patterns result from people misunderstanding the role of domains, forests, and trust relationships, including misusing domains in a forest to give regional admin teams or individual parts of the company administrative sovereignty over their part of the AD. In reality, though, every domain admin is just one step away from becoming an enterprise or domain admin in another domain of the same forest. Administrative tasks can only be segmented by assigning permissions, and the overhead for implementing this in a separate domain is identical to that for a separate organizational unit (OU). The only part where things become a little more complex is group policies, but there are solutions for that as well.

Classic mistake number two is the misuse of domains to enable individual company units to log in to "their company." A sensible naming convention for email addresses, aligning the user principal name (UPN) with these, and "Facebook login" (for example) solves this apparent problem in a far better and more sustainable way, without having to maintain a dispersed domain structure. Zero-login methods that eliminate passwords, such as smartcards, security tokens, or Windows Hello for Business, are more convenient and secure, of course. Admittedly, a problematic constellation arises in an AD forest with several or sometimes even many domains, but I will go into that in more detail in a moment.

The number of worst practices firmly anchored in daily admin work and ongoing maintenance of numerous AD environments goes far beyond the scope of this article: systems that have remained unpatched for years, insecure public key infrastructure (PKI) configurations, enterprise admin accounts for desktop management and services such as SQL or SharePoint – the list goes on and on (Figure 1). Attempts to steer day-to-day administration in a more secure direction often fail because of attitudes that say, "we can't do that," "we've always done it this way," "the boss wants it that way," and "we're too small and insignificant to be hacked." The second of the 10 Immutable Laws of Security Administration [2] from the year 2000 is apropros: Security only works if the secure way also happens to be the easy way.

Three Attack Vectors

How can it be that cybercriminals often find it easy to take control of an organization's central identity service? Aside from security vulnerabilities in Windows and other software, in most cases, attackers don't even have to deploy the big guns. They use AD features to achieve their goals.

To protect yourself against these attack techniques, you need to be clear about the benefits AD should add to your organization. Luckily, AD only has three basic functions:

- Lookup: Make information from the directory available over a defined protocol, including DNS information.
- Authentication: optionally with Kerberos or NTLM.
- Configuration of connected devices: with group policies or specific behavior on the basis of directory information, such as home drives or Windows Local Administrator Password Solution (LAPS).

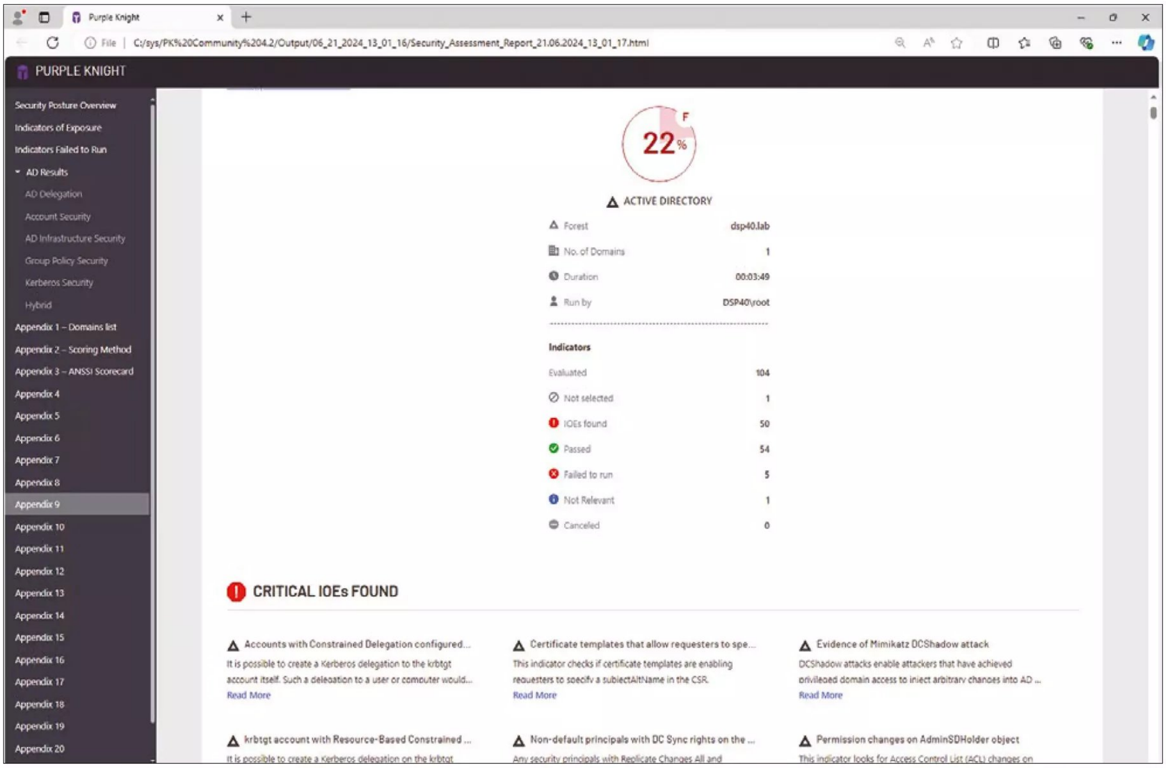
According to MITRE ATT&CK [3], attackers derive their techniques from these three AD functions and deploy them in the following important tactical fields: reconnaissance and discovery (lookup); execution (configuration); persistence, privilege escalation and lateral movement (configuration, authentication); defense evasion (configuration, authentication); and credential access (authentication). Rarely is AD used as the intial access vector or for data theft (exfiltration), and the attacker's ultimate target (impact) is more or less never that of taking over control of AD. That said, an unhardened and poorly managed AD can provide valuable service to the attacker en route to their target.

AD's functional scope does not include identity management (the only reasonably reliable item of lifecycle information about an AD object is its creation date), access management (AD provides the technological basis for authorization with groups and optionally claims, but no genuine management functions), and configuration management (not only are management functions missing here, but also the possibility to compare the current with the desired state of the environment). These points are very often misunderstood, and other products that rely on Active Directory as their data storage and delivery mechanism can offer this functionality. Again, Active Directory itself is not the management system and was never intended to be one.

Principles of Modern AD Design

To create and implement a modern AD design, you need to be aware of today's requirements and understand how these have changed for both organizations and end users over the past 20 years. In the late 1990s, when Microsoft designed Active Directory, all office workstations were hardwired and often had static IP addresses – even in larger organizations. Company sites were usually connected by lines with high latency and low bandwidth. The complete failure of a site connection for several days was an event that was to be expected not only theoretically but also practically. At the same time, all key business applications relied on centralized services such as AD and good connectivity to the back end.

Today, operators of centralized services in organizations need to meet different requirements. With many enterprise applications moving to the public cloud, the autonomy of internal systems is often less important in the event of a site connection failure. In many IT organizations, traditional AD is still the leading identity store, but most logins take place against cloud directories such as Entra ID that sync with AD. At the same time, users' work devices have become more mobile. Even at company sites, it is common practice to take a laptop to meetings in other teams' offices and conference rooms. The ability to work permanently without a connection to core systems such as Active Directory was something most organizations had to establish during the COVID-19 pandemic at the latest. Another factor that plays a far greater role today than it did 20 years ago is mergers and acquisitions or divestitures. Although companies were also acquired by other entities in the last century, and public offices were merged or specialist departments separated, it was not nearly as important then to harmonize and ultimately merge the IT systems of the merging organizations as quickly as possible.


Purple Knight
File
PURPLE KNIGHT
Security Posture Overview
Indicators of Exposure
Indicators Failed to Run
AD Results
AD Delegation
Account Security
AD Infrastructure Security
Group Policy Security
Kerberos Security
Hybrid
Appendix 1 – Domains list
Appendix 2 – Scoring Method
Appendix 3 – ANSSI Scorecard
Appendix 4
Appendix 5
Appendix 6
Appendix 7
Appendix 8
Appendix 9
Appendix 10
Appendix 11
Appendix 12
Appendix 13
Appendix 14
Appendix 15
Appendix 16
Appendix 17
Appendix 18
Appendix 19
Appendix 20
F
22%
Read More
Read More
Read More

C:/sys/PK%20Community%204.2/Output/06_21_2024_13_01_16/Security_Assessment_Report_21.06.2024_13_01_17.html

ACTIVE DIRECTORY

Forest dsp40.lab
No. of Domains 1
Duration 00:03:49
Run by DSP40\root

Indicators

Evaluated 104
Not selected 1
IOEs found 50
Passed 54
Failed to run 5
Not Relevant 1
Canceled 0

CRITICAL IOEs FOUND

Accounts with Constrained Delegation configured...
It is possible to create a Kerberos delegation to the krbtgt account itself. Such a delegation to a user or computer would...

Certificate templates that allow requesters to spe...
This indicator checks if certificate templates are enabling requesters to specify a subjectAltName in the CSR.

Evidence of Mimikatz DCShadow attack
DCShadow attacks enable attackers that have achieved privileged domain access to inject arbitrary changes into AD ...

krbtgt account with Resource-Based Constrained ...
It is possible to create a Kerberos delegation on the krbtgt

Non-default principals with DC Sync rights on the ...
Any security principals with Replicate Changes All and

Permission changes on AdminSDHolder object
This indicator looks for Access Control List (ACL) changes on

Figure 1: Security scans, like the Purple Knight scan shown here, demonstrate that the same classic mistakes keep occurring that make AD an easy target for attackers.

Experience has shown that identity and associated services such as messaging and groupware, and authorization for these and other applications, are at the forefront of this process. However, the threat situation to which central IT services are exposed has also changed fundamentally. Although firewalls, virus scanners, and encryption existed in 2000, fire and flooding were at the top of the list of specific threats to an AD infrastructure at the time, followed by hard disk failure and human error. Access to the Internet at an office or production site and an email inbox for every employee were not automatic. When professional hackers broke into a system, they usually had the specific task of stealing information. Today, the Internet, email, and instant messaging are ubiquitous, and criminals have discovered that it is far easier to get the money from the target themselves if they completely encrypt the target's data indiscriminately.

Many of the new requirements initially relate to client management and only affect AD to the extent that settings in group policies and configuration settings in AD need to be synchronized with mobile device management such as Intune. However, today's improved connectivity between locations and the relocation of some applications to the cloud offer an opportunity to streamline the design of Active Directory forests radically and therefore indirectly make them more secure.

Topology and Security Are Connected

The eighth law of the Immutable Laws of Security Administration I mentioned earlier states that the more complex the network design, the more difficult it will be to defend. In terms of Active Directory, the domain topology in a forest is a prime example of this statement. Not only does each additional domain create potential enterprise admins that security managers need to keep an eye on and limit their reach in some way, but a multidomain forest is more difficult to manage in every other aspect.

The disadvantages of this topology decision become particularly apparent if the forest is completely destroyed (e.g., by ransomware or a wiper) and needs to be restored as quickly as possible. A single-domain forest already has the entire database with the first DC restored from the backup, and only a metadata cleanup is required to restore the forest's health state. Strictly speaking, the global catalog is not necessary because it does not provide any new information compared with direct LDAP queries. In a forest with multiple domains, however, you might need to restore, clean up, and reconnect each domain to the rest of the forest. The official forest recovery manual **[4]** has now grown to 54 pages. Adding this manual to your emergency folder is a good idea, and depending on the complexity of the original forest, you can work through it in an emergency.

What used to be a very common sight with an empty forest root domain and all production objects residing in other domains seems particularly absurd in today's context when it comes to emergency recovery. Recovering the entire forest depends on successful recovery of the root domain, without it contributing anything to operations in terms of content.

Check whether a design that only contains single-domain forests is possible. Structuring by forest is an approach primarily taken because high security – Red Forest (i.e., the Enhanced Security Admin Environment, or ESAE) for classified information, research, and development, also called the admin forest, hardened forest, or the Bastion Forest – or particularly insecure (Exchange) applications need to be confined to separate forests. This approach also can be considered if business units are earmarked for divestment or if parts of the infrastructure are located in places with extremely poor connectivity so that replication and accessibility of the flexible single master operations (FSMO) roles are a problem.

A Walk in the Red Forest

You will definitely want to think about a Red Forest to secure your administrative credentials. The often-quoted statement that Microsoft classifies the ESAE model, administrative tiering, and all other components as obsolete and no longer recommends them is not entirely correct. It is simply no longer the security architecture that Microsoft would unquestioningly recommend as the standard.

The box on the front page of ESAE **[5]** is also interesting in this context. Although it does not recommend an isolated hardened forest model for most scenarios in organizations, Microsoft works with a similar architecture internally (with associated support processes and people) because of the extreme security requirements for providing trusted cloud services to companies around the world.

Like any security mechanism, IT managers need to operate a Red Forest correctly to improve the security of the entire infrastructure. This process involves setting up a unidirectional privileged access management (PAM) trust **[6]** to the Red Forest. You might also want to consider the use of smart cards for Red Forest logins and use shadow principals to assign principals from the Red Forest permissions in a Golden Forest. For further hardening, you need to restrict the permissions of Authenticated Users in the Red Forest to ensure that privileges cannot be escalated and prevent searching for and displaying objects and attributes. The trust object from the trusted forest could otherwise be exploited to get a foot in the door of the trusted forest. Windows Server 2025 closes this uncommon but easily exploitable gap.

If your AD infrastructure comprises more than one forest (including the Red Forest), you need to make sure the names of the AD sites and their assignments to the subnets match. This arrangement significantly streamlines cross-forest logins and other AD access, because the domain controller (DC) locator process starts the search for a suitable DC for the user with the site name that has already been determined for the computer from its forest. If your site and subnet topology is dispersed, you do not need to recreate it completely in every forest. You can also leave out any sites from which no logins are ever made to the forest, including the subnets assigned to them.

Address Book for Some, Reconnaissance for Others

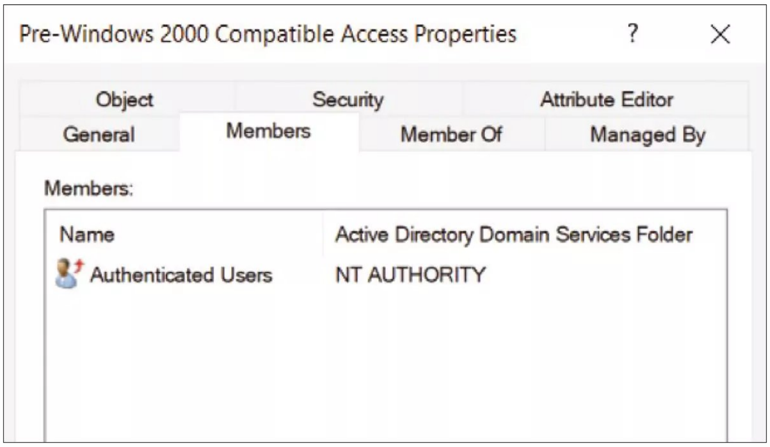
Active Directory's most important function, object lookup, is the primary goal of any LDAP directory service. Microsoft also wanted to offer a kind of enterprise address book in AD where employees could look up internal information. The search function in Windows 2000 and XP had a *Search in directory* widget to allow users to search the AD from the Start menu. This feature was rarely used because, first, users could not do much with the objects they found, and, second, widespread email and groupware products such as Exchange or Lotus Notes already offered address books at that time that went far beyond the search widget in the Start menu in terms of convenience and performance.

Moreover, these application-specific address books do not require the end user to have direct access to the LDAP directory underpinning them.

In the vast majority of cases, normal users do not need to be able to read the entire directory (domain or even forest) with the global catalog. Unfortunately, this is precisely the "view" that AD grants every member of the Authenticated Users group by default. Besides the objects themselves, the scope of attributes that can be read by default is also problematic – to the extent that it is often claimed that everyone is allowed to read everything in AD.

A large part of these excessive rights (e.g., the ability to read the `userAccountControl` attribute of another user object) is granted by the Pre-Windows 2000 Compatible Access group. After the installation of an AD domain, and even on Windows Server 2025, the Authenticated Users group is a member (Figure 2). In many organizations, removing this membership does not cause any disruptions to operations.

However, some applications require selected users to have access to certain attributes of other objects. In 99 percent of these cases, this requirement is attributable to outdated software components that still require an NT domain. In this case, you need to organize the authorizations more carefully and first remove them for Authenticated Users and Pre-Windows-2000-Compatible Access from all OUs and containers that contain privileged objects; then, grant these permissions to any admin groups necessary, if you have not already done so. No normal user needs to be able to list administrators, highly privileged groups, and Tier 0 systems. Also, consider the permissions of the `AdminSDHolder` container.


Pre-Windows 2000 Compatible Access Properties

Members:

Name	Active Directory Domain Services Folder
Authenticated Users	NT AUTHORITY

Figure 2: The Pre-Windows 2000 Compatible Access group gives users substantial read authorizations and should be avoided.

The second step is to clear out the Pre-Windows 2000 Compatible Access group. If it turns out that some applications still require permissions granted by it, you can add the required machines or users again. If it unexpectedly turns out that an application still needs these authorizations for all standard users, you can set up a special group to grant them, and you can then delete the group in one fell swoop after you have finally removed the obsolete applications from your AD.

The third measure is to restrict attribute visibility further for authenticated users. You will also want to look at the AD configuration partition, which again contains much information that is of interest to a potential attacker but is unimportant to a normal user.

However, be careful: The Windows Network Policy Server role needs some of the permissions granted by the Pre-Windows 2000 Compatible Access group, so you will need to grant the required read permissions to the machines involved.

Try to work exclusively with *Allow* permissions, which do not just apply to permissions in Active Directory, but to all authorizations in Windows. If this route is not possible and you need to go for *Deny* instead, you can use the list object mode to control visibility [7] in a granular way. Note that in large environments with deep OU hierarchies, enabling list object mode will generate a noticeable additional load on the domain controllers.

Although restricting visibility effectively impedes initial reconnaissance by cybercriminals, it can also be used against you as a defensive evasion tactic in a later phase of the attack. The attacker will be able to create a highly privileged user and revoke the access rights to this object from all AD principals. This action does not prevent the account from being authenticated and authorized, but it makes the account invisible to normal consoles and scripts. However, some specialist tools [8] can be found that detect hidden admins.

If you use tools in your AD that evaluate the objects and their configurations to assess the attack surface, you need to make sure your hardening activities do not restrict the capabilities of these tools. If necessary, set up a special account for this purpose and authorize it to see everything.

Conclusions

Although many believe the Active Directory service is inherently insecure, I talked about how to achieve a secure AD design by showing where the security pitfalls lie and how you can avoid them. In the next article in this series, I will look at how to harden the authentication process in Active Directory to prevent credential theft, before moving on to look at configuration maintenance and the practical implementation of these strategies in an existing infrastructure. ■

Info

[1] Microsoft Purview: [https://learn.microsoft.com/en-us/purview/information-protection]

[2] Immutable Laws of Security Administration: [https://web.archive.org/web/20050314224039/http://www.microsoft.com/technet/archive/community/columns/security/essays/10salaws.mspx]

[3] MITRE ATT&CK: [https://attack.mitre.org]

[4] Guide for restoring the AD forest: [https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/forest-recovery-guide/ad-forest-recovery-guide]

[5] Advanced security admin environment: [https://learn.microsoft.com/en-us/security/privileged-access-workstations/esae-retirement]

[6] Privileged Access Management and Shadow Principals: [https://www.teal-consulting.de/en/2018/08/14/esae-series-part-3-privileged-access-management-and-the-shadow-principal-feature/]

[7] Managing object visibility: [https://learn.microsoft.com/en-us/windows/win32/ad/controlling-object-visibility]

[8] Finding hidden AD objects: [https://www.semperis.com/blog/find-hidden-active-directory-objects/]

Author

Evgenij Smirnov has been working with computers since the age of five and delivering IT solutions for almost 30 years. His Active Directory and Exchange background naturally led to PowerShell, of which he's been an avid user and proponent since its first release. Evgenij is an active community lead at home in Berlin, a leading contributor to German online communities, and an experienced user group and conference speaker. He has been a PowerShell MVP since 2020. He is also the author of the 2024 book *Building Modern Active Directory*.

shop.linuxnewmedia.com/shop

Discover the past and invest in a new year of IT solutions at Linux New Media's online store.

Want to subscribe?

Searching for that back issue you really wish you'd picked up at the newsstand?

shop.linuxnewmedia.com/shop

DIGITAL & PRINT SUBSCRIPTIONS

ADMIN Network & Security
Monitoring
LINUX MAGAZINE
What Comes After Git?

SPECIAL EDITIONS

GETTING STARTED WITH LINUX
LibreOffice Expert
LINUX SHELL HANDBOOK
Cool Linux Hacks
MakerSpace
THE BEST OF LINUX MAGAZINE

Processing the new sudo logging format

Neat Packaging

We present an example that shows how syslog-ng uses the new logging format of the sudo tool to process sudo messages. By Thorsten Scherf

The Linux sudo command helps Linux administrators run specific programs in the context of another user, typically root. In this way, non-privileged users can run management programs on a system. The tool writes to a log to keep track of users who used sudo to run programs. The default format has been plain text, but from version 1.9.4, it can be configured to log in JSON format.

sudo Logging

By default, sudo uses syslog to send log messages, which means that, on a Linux system, the messages end up in the system journal. You can choose between JSON and sudo plain text as the log format. Text ensures that the messages are easy to read while keeping the amount of information that ends up in the log to a minimum (Figure 1).

If you are looking for more information on a sudo log event, you can change the format of the log message to JSON, which makes several data fields available in the journal (Figure 2). You will find more detailed information on the individual fields of the log format by entering

```
man 5 sudoers
```

for the sudoers help page.

Logging Child Processes

Starting in version 1.9.8, a sudo option lets you log the child processes of the application launched by sudo [1]. For example, if you authorize a user to launch an editor with sudo, you would normally only see the call to the editor command in the logfile. However, if the user calls other commands from inside the editor, these calls are not shown in the logs. The case in which you potentially give a user the right to launch a new shell with sudo to gain complete root access to the system is particularly interesting. In the logs, you would only see the call to the shell, but not which tools were launched in the shell session.

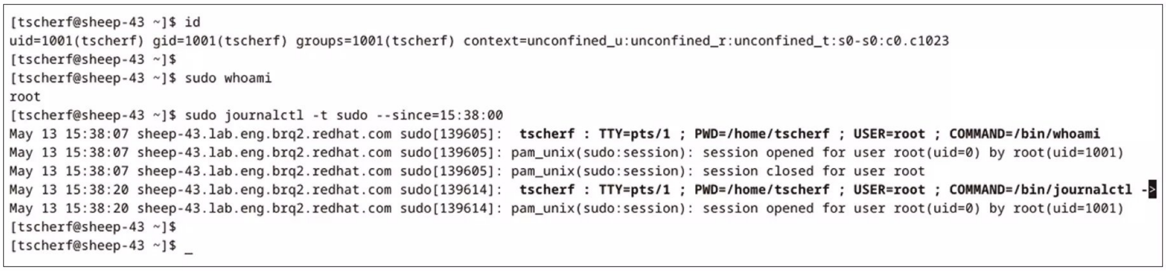
To log all commands of a sudo session, you can use the `log_subcmds` option within the sudoers file. Of course, a large number of log messages will then end up in the journal, which will be almost impossible to analyze manually, and you will very likely overlook information that could help you track specific actions on a system.

Central Log

To solve this problem, it makes sense to collect all log messages from the individual computers on a central system, where they can then be managed and analyzed with your preferred log management software, such as Elasticsearch. Luckily, sudo can also store the log entries in a regular file instead of routing them by syslog. You can then use the machine-readable JSON format to process the sudo messages automatically.

However, sudo previously used a human-readable JSON format when writing messages to a file:

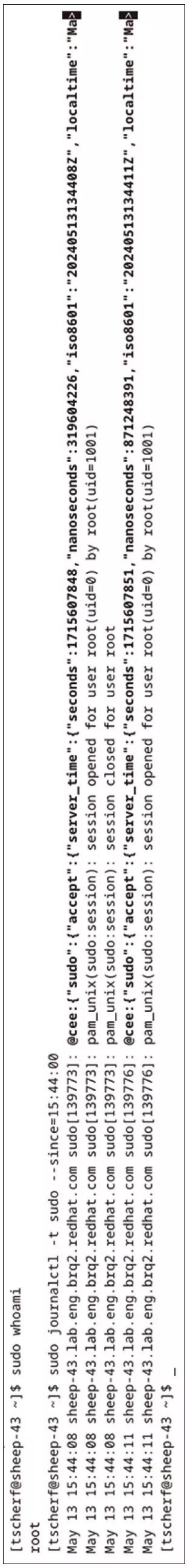
```
{
  "accept": {
    "server_time": {
      "seconds": 1716370731,
      "nanoseconds": 137298080,
      "iso8601": "20240522093851Z",
      "localtime": "May 22 09:38:51"
    },
    [...]
  },
}
```



```
[tscherf@sheep-43 ~]$ id
uid=1001(tscherf) gid=1001(tscherf) groups=1001(tscherf) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[tscherf@sheep-43 ~]$
[tscherf@sheep-43 ~]$ sudo whoami
root
[tscherf@sheep-43 ~]$ sudo journalctl -t sudo --since=15:38:00
May 13 15:38:07 sheep-43.lab.eng.brq2.redhat.com sudo[139605]:  tscherf : TTY=pts/1 ; PWD=/home/tscherf ; USER=root ; COMMAND=/bin/whoami
May 13 15:38:07 sheep-43.lab.eng.brq2.redhat.com sudo[139605]: pam_unix(sudo:session): session opened for user root(uid=0) by root(uid=1001)
May 13 15:38:07 sheep-43.lab.eng.brq2.redhat.com sudo[139605]: pam_unix(sudo:session): session closed for user root
May 13 15:38:20 sheep-43.lab.eng.brq2.redhat.com sudo[139614]:  tscherf : TTY=pts/1 ; PWD=/home/tscherf ; USER=root ; COMMAND=/bin/journalctl ->
May 13 15:38:20 sheep-43.lab.eng.brq2.redhat.com sudo[139614]: pam_unix(sudo:session): session opened for user root(uid=0) by root(uid=1001)
[tscherf@sheep-43 ~]$
[tscherf@sheep-43 ~]$ _
```

Figure 1: By default, sudo uses an easy-to-read plain text log format for all messages routed by syslog.

Photo by Kelly Sikkema on Unsplash



```
[tscherf@sheep-43 ~]$ sudo whoami
root
[tscherf@sheep-43 ~]$ sudo journalctl -t sudo --since=15:44:00
May 13 15:44:08 sheep-43.lab.eng.brq2.redhat.com sudo[139773]: @cee:{"sudo":{"accept":{"server_time":{"seconds":1715607848,"nanoseconds":319604226,"iso8601":"20240513134408Z","localtime":"Ma>
May 13 15:44:08 sheep-43.lab.eng.brq2.redhat.com sudo[139773]: pam_unix(sudo:session): session opened for user root(uid=0) by root(uid=1001)
May 13 15:44:08 sheep-43.lab.eng.brq2.redhat.com sudo[139773]: pam_unix(sudo:session): session closed for user root
May 13 15:44:11 sheep-43.lab.eng.brq2.redhat.com sudo[139776]: @cee:{"sudo":{"accept":{"server_time":{"seconds":1715607851,"nanoseconds":871248391,"iso8601":"20240513134411Z","localtime":"Ma>
May 13 15:44:11 sheep-43.lab.eng.brq2.redhat.com sudo[139776]: pam_unix(sudo:session): session opened for user root(uid=0) by root(uid=1001)
[tscherf@sheep-43 ~]$ _
```

Figure 2: Log messages in JSON format provide more information.

This file is easy to read because the log message comprises several lines of data fields, which facilitates finding the desired information. However, you might prefer sudo to output the same compact JSON format used when you route the messages to the local journal by syslog, and this is precisely what you can do with the current sudo 1.9.16 version.

Compact JSON for Logfiles

This new format not only saves space but also makes it easier to process the logfiles automatically. Besides the sudo log format, it offers two new formats: `json_pretty` and `json_compact`. Whereas `json_pretty` is identical to the old JSON format, `json_compact` crams the log messages in JSON format into a single line, much like logging with syslog. You can then either process these locally with a JSON parser (e.g., `jq`) or forward them to a central logging instance for further analysis.

In the following example, the objective is to evaluate all the logs from a local syslog-ng instance. Alternatively, you could also send the logs from there to some other software or to a central log server. The example is merely intended to illustrate the options the new log format allows. To customize the sudo configuration, you would traditionally call up the `visudo` tool, which gives you an exclusive lock on the file and prevents any parallel changes to the file from overwriting your settings. Alternatively, you could create a new file in the `/etc/sudoers.d/` folder for your own customizations. To edit this file, simply call:

```
# visudo -f /etc/sudoers.d/logging
```

In the file, specify that sudo should only write log messages to a text file and not additionally send them by syslog to the local journal. Now enable the new `json_compact` format. In this example, I'm assuming you also want to log calls to all child processes of the permitted sudo commands:

```
Defaults !syslog
Defaults logfile=/var/log/sudo
Defaults log_format=json_compact
Defaults log_subcmds
```

Listing 1 shows a simple syslog-ng configuration file that draws on syslog-ng to evaluate the JSON-formatted `/var/log/sudo` logfile and convert the logs to a different format to generate the new `/var/log/sudo-text` logfile. The results are shown in Listing 2. Of course, you could simply forward the JSON-formatted logfile to a different logging instance to analyze the logs there. The syslog-ng website provides some examples [2].

Listing 1: `/etc/syslog-ng/conf.d/sudo.conf`

```
# cat sudo.conf
source s_sudojson {
   file("/var/log/sudo" flags(no-parse));
};
parser p_json {
     json-parser();
};
destination d_sudo-welf {
   file("/var/log/sudo"
     template("$(format-welf --scope nv_pairs --exclude
         MESSAGE --exclude accept.submitenv)\n\n")
   );
   file("/var/log/sudo-text"
     template("${DATE} user ${accept.submituser} ran
        ${accept.command} on host ${HOST} using sudo\n")
   );
};
log {
   source(s_sudojson);
   parser(p_json);
   destination(d_sudo);
};
```

Listing 2: New Logfile Format

```
May 22 09:38:51 user tscherf ran /usr/bin/vim on host
   master.ipa.test using sudo
May 22 09:38:52 user tscherf ran /usr/bin/who on host
   master.ipa.test using sudo
```

Conclusions

Thanks to the new sudo log format `json_compact`, you can easily generate machine-readable sudo logfiles, which you can then forward to a log management system for analysis downstream. If you prefer to inspect the messages manually instead, you can either use the sudo format or `json_pretty` if you need more information. ■

Info

[1] Logging sudo child processes: [https://www.sudo.ws/posts/2021/08/what-is-coming-in-sudo-1.9.8/]

[2] syslog-ng configuration examples: [https://www.syslog-ng.com/community/b/blog/posts/type-support-working-with-sudo-logs-in-syslog-ng-4-0]

The Author

Thorsten Scherf is the global Product Lead for Identity Management and Platform Security in Red Hat's Product Operations group. He is a regular speaker at various international conferences and writes a lot about open source software.

Hone Your Skills with Special Issues!

Get to know Shell, LibreOffice, Linux, and more from our Special Issues library.

The *Linux Magazine* team has created a series of single volumes that give you a deep-dive into the topics you want.

Available in print or digital format.

Check out the full library: shop.linuxnewmedia.com/shop

image © roystudio, 123RF.com

BUILD A RASP PI RADIO!
MakerSpace #02
RASPBERRY PI GEEK
THE COMPLETE ARCHIVE
2,000 pages of maker projects and more!
FREE DVD $39.90 VALUE!
ARCHIVE DVD
MakerSpace
RASP PI TRICKS
TRY ARTIFICIAL INTELLIGENCE ON A PI
MakerSpace #04
BUILD INFO DISPLAYS
HANDS-ON PROJECTS FOR MAKERS
Retro Computing
C64x with Commodore OS
FPGA-based Spectrum Next
Smart Home
Automate your house
Check plants and
Low-Code
Programmi
Learn Node-RED
then add Pytho
LINUX NEW MEDIA
RASP PI TRICKS
FEED YOUR FISH WHILE YOU'RE AWAY
MakerSpace #03
HANDS-ON PROJEC
FOR MAKE
Solar-Powered IoT
void the battery dance with
sun-powered Pi Pico
igation Innovation
er your plants without
ing wet
FIND THE PERFECT TOOL FOR THE TASK
2024 EDITION
Cool Linux Hacks
Tricks and shortcuts for Linux geeks
• Run macOS in a Virtual Machine on Your Linux Desktop
• Let AI Help You with Shell Commands
114 HACKS
LINUX Special
301 BEST BASH COMMANDS
LINUX SHELL HANDBOOK
2025 Edition
SUPERCHARGE YOUR LINUX SKILLS
Power at Your
LibreOffice Expert
DVD INSIDE!
LibreOffice Full Version
Dive deep into the world's greatest free office suite
2024/25 Edition
LibreOffice Expert
Edit and Save MS Office Files
Replace MS Office and Google Docs!
LibreOffice
Includes full versions for Windows, macOS, and Linux
FREE DVD!
openSUSE Leap 15.5
JOIN THE LINUX REVOLUTION!
ALL THE SOFTWARE YOU NEED!
GETTING STARTED WITH LINU
MORE POWERFUL • MORE
LEARN HOW TO SET UP
• Listen to Music • Play G
• Surf the Web •
ST
LINUX NEW MEDIA
WWW.LINUX-MAGAZINE.C
THE BEST OF
LINUX MAGAZINE
2024
Advanced Techniques from Linux Experts:
• Smoother operation
• Better security
• Smarter tools
Zint
Conjure up your own barcodes
Node-RED
Coding without the code
Shell Genie
Convert your natural-language questions to Linux commands
ChatGPT Clients
AI at the Linux command line
Photo Collages
Use these Linux tools to mash up your photos
Plus much more inside!
Check out our best articles from 2024
LINUX NEW MEDIA
WWW.LINUX-MAGAZINE.COM

ADMIN
Network & Security

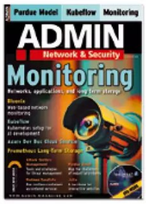
NEWSSTAND

Order online:
bit.ly/ADMIN-Library

ADMIN is your source for technical solutions to real-world problems. Every issue is packed with practical articles on the topics you need, such as: security, cloud computing, DevOps, HPC, storage, and more! Explore our full catalog of back issues for specific topics or to complete your collection.

#85 – January/February 2025

Monitoring

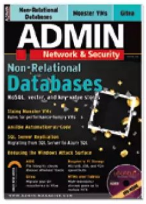


We look at infrastructure and systems monitoring and evaluate the tools needed to scale or store data in the long term for analysis.

On the DVD: Fedora Server 41

#84 – November/December 2024

Non-Relational Databases

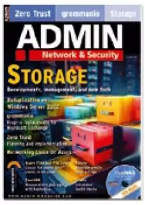


NoSQL databases manage data unsuitable for table-based relational databases and deliver efficiencies in scaling and high availability.

On the DVD: Ubuntu 24.04 LTS Server "Noble Numbat"

#83 – September/October 2024

Storage

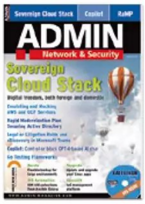


Improved management, suitable drivers, standardized protocol structures, and advancements in future-proof hardware and software lead to more durable, more manageable, and easier-to-repair storage.

On the DVD: TrueNAS SCALE 24.04 "Dragonfish"

#82 – July/August 2024

Sovereign Cloud Stack

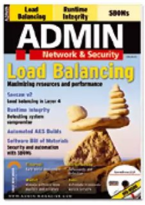


SCS liberates your data centers from monopolistic operations and companies beholden to out-country laws and regulations.

On the DVD: Kali Linux 2024.2

#81 – May/June 2024

Load Balancing

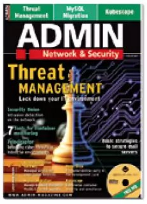


Load balancing on heavily frequented networks improves performance, availability, security, scalability, and the ability to handle peak loads.

On the DVD: SystemRescue 11.01

#80 – March/April 2024

Threat Management



Digital infrastructures are vulnerable to all kinds of attacks. You need strategies and tools to detect and defend.

On the DVD: openSUSE Leap 15.5

WRITE FOR US

Admin: Network and Security is looking for good, practical articles on system administration topics. We love to hear from IT professionals who have discovered innovative tools or techniques for solving real-world problems.

Tell us about your favorite:

- Interoperability solutions
- Practical tools for cloud environments
- Security problems and how you solved them
- Ingenious custom scripts
- Unheralded open source utilities
- Windows networking techniques that aren't explained (or aren't explained well) in the standard documentation

We need concrete, fully developed solutions: installation steps, configuration files, examples – we are looking for a complete discussion, not just a "hot tip" that leaves the details to the reader.

If you have an idea for an article, send a 1-2 paragraph proposal describing your topic to: *edit@admin-magazine.com*.

Authors

Marcin Gastol	26
Mark Heitbrink	80
Ken Hess	3
Martin Gerhard Loschwitz	72
Dr. Holger Reibold	42, 50, 54
Daniel Richey	10
Thorsten Scherf	48, 62, 78, 92
Abe Sharp	30
Evgenij Smirnov	86
Andreas Stolzenberger	16
Matthias Wübbeling	58

Contact Info

Editor in Chief
Joe Casad, jcasad@linuxnewmedia.com

Managing Editors
Rita L Sooby, rsooby@linuxnewmedia.com
Lori White, lwhite@linuxnewmedia.com

Senior Editor
Ken Hess

Localization & Translation
Ian Travis

News Editor
Amber Ankerholz

Copy Editors
Amy Pettle, Aubrey Vaughn

Layout
Dena Friesen, Lori White

Cover Design
Lori White, Illustration based on graphics by archman, 123RF.com

Advertising
Jessica Pryor, jpryor@linuxnewmedia.com

Publisher
Brian Osborn

Marketing Communications
Gwen Clark, gclark@linuxnewmedia.com
Linux New Media USA, LLC
4840 Bob Billings Parkway, Ste 104
Lawrence, KS 66049 USA

Customer Service / Subscription
For USA and Canada:
Email: cs@linuxnewmedia.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)

For all other countries:
Email: subs@linuxnewmedia.com
www.admin-magazine.com

While every care has been taken in the content of the magazine, the publishers cannot be held responsible for the accuracy of the information contained within it or any consequences arising from the use of it. The use of the DVD provided with the magazine or any material provided on it is at your own risk.

Copyright and Trademarks © 2025 Linux New Media USA, LLC.

No material may be reproduced in any form whatsoever in whole or in part without the written permission of the publishers. It is assumed that all correspondence sent, for example, letters, email, faxes, photographs, articles, drawings, are supplied for publication or license to third parties on a non-exclusive worldwide basis by Linux New Media unless otherwise stated in writing.

All brand or product names are trademarks of their respective owners. Contact us if we haven't credited your copyright; we will always correct any oversight.

Printed in Nuremberg, Germany by be1druckt GmbH.

Distributed by Seymour Distribution Ltd, United Kingdom

ADMIN (Print ISSN: 2045-0702, Online ISSN: 2831-9583, USPS No: 347-931) is published bimonthly by Linux New Media USA, LLC, and distributed in the USA by Asendia USA, 701 Ashland Ave, Folcroft PA. March/April 2025. Application to Mail at Periodicals Postage Prices is pending at Philadelphia, PA and additional mailing offices. POSTMASTER: send address changes to Linux Magazine, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA.

Represented in Europe and other territories by: Sparkhaus Media GmbH, Bialasstr. 1a, 85625 Glonn, Germany.

ADMIN 87

Available Starting June 6

Our next issue will be packed with all the great content you expect from *ADMIN*. Here are a few of the upcoming articles:

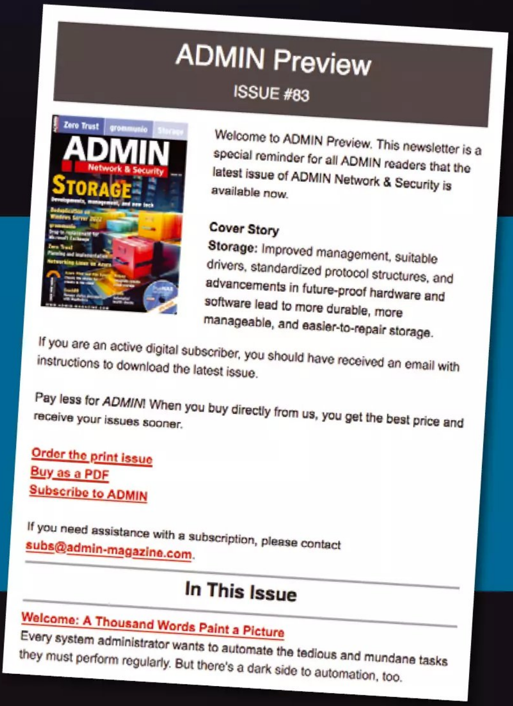
- Visualizing File Access Patterns
- NAPS2 SDK
- OCR Tools
- Netdata
- And much more!

Please note: Articles could change before the next issue.

BE THE FIRST TO SEE WHAT'S NEXT

Subscribe free to the *ADMIN* Preview newsletter and get a sneak peek at every article included in the next issue of *ADMIN*.

Sign up today at https://bit.ly/admin-preview



ADMIN Preview

ISSUE #83

Welcome to ADMIN Preview. This newsletter is a special reminder for all ADMIN readers that the latest issue of ADMIN Network & Security is available now.

Cover Story

Storage: Improved management, suitable drivers, standardized protocol structures, and advancements in future-proof hardware and software lead to more durable, more manageable, and easier-to-repair storage.

If you are an active digital subscriber, you should have received an email with instructions to download the latest issue.

Pay less for *ADMIN*! When you buy directly from us, you get the best price and receive your issues sooner.

Order the print issue
Buy as a PDF
Subscribe to ADMIN

If you need assistance with a subscription, please contact subs@admin-magazine.com.

In This Issue

Welcome: A Thousand Words Paint a Picture

Every system administrator wants to automate the tedious and mundane tasks they must perform regularly. But there's a dark side to automation, too.

Image © artnovielysa, 123RF.com

ISC High Performance
The HPC Event.

CONNECTING THE DOTS ↘

40 YEARS ISC

JOIN US
IN HAMBURG

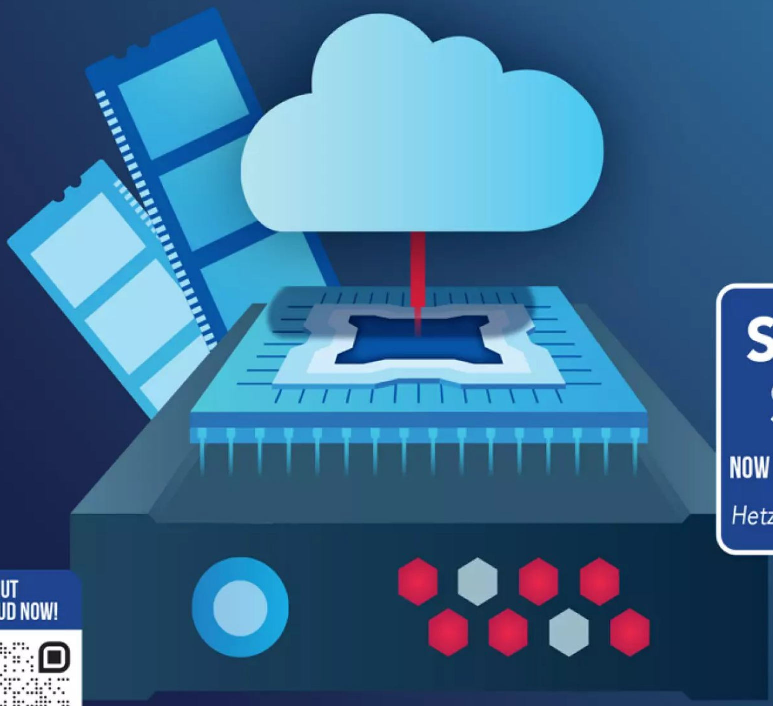
ISC HIGH PERFORMANCE 2025
JUNE 10–13 | HAMBURG, GERMANY

THE EVENT FOR HIGH PERFORMANCE COMPUTING, ARTIFICIAL INTELLIGENCE, DATA ANALYTICS & QUANTUM COMPUTING

Join us in celebrating a milestone: 40 incredible years of connecting communities and technologies. An exhilarating program with unique formats awaits 3,500 attendees. Let's make memories together!

Early-bird registration opens on March 20 and ends on April 23, 2025.

isc-hpc.com



HETZNER

PURE POWER AT
INCREDIBLY LOW PRICES

SAVE
$10
NOW WITH THE CODE:
HetznerXAdmin10

CHECK OUT
HETZNER CLOUD NOW!

CLOUD SERVER CPX11

- ✓ 2 x **shared** vCPU
  AMD EPYC™ 7002
- ✓ 2 GB RAM
- ✓ 40 GB NVMe SSD
- ✓ Location Germany, Finland, USA & Singapore
- ✓ Hourly billing
- ✓ No minimum contract

monthly from:
$ 4.49 | € 3.85

CLOUD SERVER CCX13

- ✓ 2x **dedicated** vCPU
  AMD Milan EPYC™ 7003 | Genoa EPYC™ 9654
- ✓ 8 GB RAM
- ✓ 80 GB NVMe SSD
- ✓ Location Germany, Finland, USA & Singapore
- ✓ Hourly billing
- ✓ No minimum contract

monthly from:
$ 13.49 | € 11.99

Credit code worth $10 / €10 - only for new customers - activate before 06/30/26 - valid for 3 months after activation. All prices exclude VAT and are subject to the terms and conditions of Hetzner Online GmbH. Prices are subject to change. All rights reserved by the respective manufacturers.

cloud.hetzner.com